```
DDDDDDDDDDDD        IIIIIIIII     FFFFFFFFFFFFFFF
DDDDDDDDDDDD        IIIIIIIII     FFFFFFFFFFFFFFF
DDDDDDDDDDDD        IIIIIIIII     FFFFFFFFFFFFFFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFFFFFFFFFFF
DDD          DDD       III       FFFFFFFFFFFF
DDD          DDD       III       FFFFFFFFFFFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDD          DDD       III       FFF
DDDDDDDDDDDD        IIIIIIIII     FFF
DDDDDDDDDDDD        IIIIIIIII     FFF
DDDDDDDDDDDD        IIIIIIIII     FFF
```

FILEID**OUTPUT

```
  OOOOOO    UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT
  OOOOOO    UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT
OO      OO  UU      UU      TT      PPPPPPPP    UU      UU      TT
OO      OO  UU      UU      TT      PPPPPPPP    UU      UU      TT
OO      OO  UU      UU      TT      PP          UU      UU      TT
OO      OO  UU      UU      TT      PP          UU      UU      TT
OO      OO  UU      UU      TT      PP          UU      UU      TT      ....
OO      OO  UU      UU      TT      PP          UU      UU      TT      ....
  OOOOOO    UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT      ....
  OOOOOO    UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

DIR
VO4

: F

```
;     1          0001  0 MODULE DIF_OUTPUT (                    ! Differences output routines
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 ADDRESSING_MODE (EXTERNAL=GENERAL,
;     4          0004  0                                  NONEXTERNAL=LONG_RELATIVE),
;     5          0005  0                 IDENT = 'V04-000'
;     6          0006  0                 ) =
;     7          0007  1 BEGIN
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:     DCL Differences command
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !                The DCL DIFFERENCES command compares the contents of
;    37          0037  1 !                two files.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !                VAX native, user mode
;    41          0041  1 !
;    42          0042  1 !--
;    43          0043  1 !
;    44          0044  1 ! AUTHOR:       Peter George, Benn Schreiber    CREATION DATE:  1-August-1981
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 !       V03-002 PCG0002         Peter George            11-Apr-1984
;    49          0049  1 !               Fix ACCVIO from DIFF/MERGED=0.
;    50          0050  1 !
;    51          0051  1 !       V03-001 PCG0001         Peter George            13-Oct-1982
;    52          0052  1 !               Skip matches that preceed a merged output section.
;    53          0053  1 !               Do not put change bars in front of ignored headers.
;    54          0054  1 !               Never output ignored records.
;    55          0055  1 !--
;    56          0056  1
```

```
;    58          0057  1 LIBRARY
;    59          0058  1     'SYS$LIBRARY:STARLET.L32';
;    60          0059  1
;    61          0060  1 REQUIRE
;    62          0061  1         'DIFPRE';                          ! DIF prefix file
;    63          0135  1 REQUIRE
;    64          0136  1         'DIFDEF';                          ! DIF data structures
;    65          0367  1
;    66          0368  1 !
;    67          0369  1 ! Difference global data
;    68          0370  1 !
;    69          0371  1 EXTERNAL
;    70          0372  1     dif$gl_commdesc : BBLOCK,              ! Desc for buffer of comment delimiters
;    71          0373  1     dif$gl_cmdesc : BBLOCK,                ! Descriptor for command line
;    72          0374  1     dif$gl_ignore : BBLOCK,                ! Flags of characters to ignore
;    73          0375  1     dif$gl_header,                         ! No. of lines to skip as header
;    74          0376  1     dif$gl_match,                          ! No. of records that constitute a match
;    75          0377  1     dif$gl_maxdif,                         ! Maximum number of unmatched records
;    76          0378  1     dif$gl_merged,                         ! No. of matched lines to follow each list of differences
;    77          0379  1     dif$gl_parallel,                       ! Same as above for parallel
;    78          0380  1     dif$gl_wndwsiz,                        ! No. of records to search before
;    79          0381  1                                            ! declaring a mismatch
;    80          0382  1     dif$gl_flags : BBLOCK,                 ! Flags
;    81          0383  1     dif$gl_difrec,                         ! No. of difference records found
;    82          0384  1     dif$gl_difsec,                         ! No. of difference sections detected
;    83          0385  1     dif$gl_dumpwidth,                      ! Width of hex/octal data part of line
;    84          0386  1     dif$gl_entsperline,                    ! No. of entries on a hex/octal line
;    85          0387  1     dif$gl_width,                          ! Width of lines in output listing
;    86          0388  1     dif$gl_parwidth,                       ! Width of lines in parallel listing
;    87          0389  1     dif$gl_inbuf,                          ! Address of the input record buffer
;    88          0390  1     dif$gl_outbuf : REF VECTOR [, BYTE],   ! Address of the output record buffer
;    89          0391  1     dif$gl_faofullbuf : BBLOCK,            ! Hex/octal fao full line buffer
;    90          0392  1     dif$gl_faopartbuf : BBLOCK,            ! Hex/octal fao partial line buffer
;    91          0393  1     dif$gl_faofulldesc : BBLOCK,           ! String desc for hex/octal full output line
;    92          0394  1     dif$gl_faopartdesc : BBLOCK,           ! String desc for hex/octal partial output line
;    93          0395  1 !
;    94          0396  1 ! Input and output file data structures
;    95          0397  1 !
;    96          0398  1     dif$gl_masdesc : BBLOCK,               ! String desc for input file
;    97          0399  1     dif$gl_masfdb : BBLOCK,                ! Master file fdb
;    98          0400  1     dif$gl_masrab : BBLOCK,                ! RAB for master file
;    99          0401  1     dif$gl_revdesc : BBLOCK,               ! String desc for revision file
;   100          0402  1     dif$gl_revfdb : BBLOCK,                ! Revision file fdb
;   101          0403  1     dif$gl_revrab : BBLOCK,                ! RAB for revision file
;   102          0404  1     dif$gl_outdesc : BBLOCK,               ! String desc for output file
;   103          0405  1     dif$gl_outrab : BBLOCK;                ! RAB for output file
;   104          0406  1
;   105          0407  1 EXTERNAL ROUTINE
;   106          0408  1     dif$format_hex_octal,                  ! Format a record in either hex or octal
;   107          0409  1     ots$cvt_l_ti,                          ! Convert longword to text
;   108          0410  1     sys$fao;                               ! FAO conversion routine
;   109          0411  1
;   110          0412  1 FORWARD ROUTINE
;   111          0413  1     additional_output,                     ! Output 2nd, 3rd, ... listings
;   112          0414  1     get_rfa_text,                          ! Get record text using RFA
;   113          0415  1     init_hex_octal,                        ! Prepare for hex or octal output
;   114          0416  1     insert_linenum,                        ! Insert a line # in output line
```

```
;  115          0417  1      output_changebar,                       ! Output in change bar format
;  116          0418  1      output_listing_trailer,                 ! Output listing trailer
;  117          0419  1      output_cmdfile,                         ! Output file line of trailer
;  118          0420  1      output_cmdfao,                          ! Output command using $FAO
;  119          0421  1      output_cmdcounted,                      ! Output counted entity
;  120          0422  1      output_cmdentity,                       ! Output command line entity
;  121          0423  1      output_merged,                          ! Output in merged format
;  122          0424  1      output_parallel,                        ! Output in parallel format
;  123          0425  1      output_separated,                       ! Output in separated format
;  124          0426  1      output_slp,                             ! Output in SLP format
;  125          0427  1      put_blank,                              ! Output a blank line
;  126          0428  1      put_desc,                               ! Output a descriptor
;  127          0429  1      put_hex_octal_header,                   ! Output hex/octal record header
;  128          0430  1      put_idline,                             ! Output a file id line
;  129          0431  1      put_parallel_idline,                    ! Output a parallel file id line
;  130          0432  1      put_record,                             ! Output a record in appropriate radix
;  131          0433  1      put_record_ascii,                       ! Output an ascii record
;  132          0434  1      put_record_hex_octal,                   ! Output a hex or octal record
;  133          0435  1      put_record_parallel,                    ! Output a parallel format record
;  134          0436  1      translate_tabs,                         ! Convert tabs to blanks
;  135          0437  1      write_mismatch;                         ! Output records in a mismatch
;  136          0438  1
;  137          0439  1  OWN
;  138          0440  1      cmd_bufpos,                             ! Position for command output
;  139          0441  1      cr :          COUNTEDSTRING ('<CR>'),
;  140          0442  1      ff :          COUNTEDSTRING ('<FF>'),
;  141          0443  1      lf :          COUNTEDSTRING ('<LF>'),
;  142          0444  1      vt :          COUNTEDSTRING ('<VT>'),
;  143          0445  1      blanks :      COUNTEDSTRING ('          '),
;  144          0446  1      period :      COUNTEDSTRING ('.'),
;  145          0447  1      change :      COUNTEDSTRING (' ***CHANGE***'),
;  146          0448  1      difrec :      COUNTEDSTRING ('Number of difference records found: !ZL'),
;  147          0449  1      difsec :      COUNTEDSTRING ('Number of difference sections found: !ZL'),
;  148          0450  1      file :        COUNTEDSTRING ('File '),
;  149          0451  1      hexfull :     COUNTEDSTRING ('!!!ZL(9XL) !!!ZLAF !!6XL'),
;  150          0452  1      hexheader :   COUNTEDSTRING ('RECORD NUMBER !ZL (!8XL) LENGTH !ZL (!8XL) !AS'),
;  151          0453  1      hexpart :     COUNTEDSTRING ('!!!!!!ZL(9XL) !!!!!ZLAF !!!!6XL'),
;  152          0454  1      octfull :     COUNTEDSTRING ('!!!ZL(12OL) !!!ZLAF !!6OL'),
;  153          0455  1      octheader :   COUNTEDSTRING ('RECORD NUMBER !ZL (!8OL) LENGTH !ZL (!8OL) !AS'),
;  154          0456  1      octpart :     COUNTEDSTRING ('!!!!!!ZL(12OL) !!!!!ZLAF !!!!6OL'),
;  155          0457  1      slpoprs :     COUNTEDSTRING ('-\%a/<'),
;  156          0458  1      stars :       COUNTEDSTRING ('************');
;  157          0459  1
;  158          0460  1  LITERAL
;  159          0461  1      ffeed = 12;                             ! form feed character
;  160          0462  1
;  161          0463  1  EXTERNAL LITERAL
;  162          0464  1      dif$_readerr,
;  163          0465  1      dif$_writeerr;
;  164          0466  1
```

```
;  166     0467  1 GLOBAL ROUTINE write_mismatch =
;  167     0468  2 BEGIN
;  168     0469  2
;  169     0470  2 !++
;  170     0471  2 !
;  171     0472  2 ! FUNCTIONAL DESCRIPTION:
;  172     0473  2 !
;  173     0474  2 !       When the end of a mismatch is detected, this routine is used t
;  174     0475  2 !       call the output routine for the appropriate user-specified output
;  175     0476  2 !       format.  The order of the IF - THEN - ELSE statements is significant
;  176     0477  2 !       because this control structure is imbedded in several other places
;  177     0478  2 !       in the Diff code.
;  178     0479  2 !
;  179     0480  2 ! INPUTS:
;  180     0481  2 !
;  181     0482  2 !       None
;  182     0483  2 !
;  183     0484  2 ! OUTPUTS:
;  184     0485  2 !
;  185     0486  2 !       dif$gl_difsec is incremented
;  186     0487  2 !
;  187     0488  2 ! ROUTINE VALUES:
;  188     0489  2 !
;  189     0490  2 !       Always true
;  190     0491  2 !
;  191     0492  2 !--
;  192     0493  2 dif$gl_difsec = .dif$gl_difsec + 1;             ! Incr count of diff sections
;  193     0494  2
;  194     0495  2 IF .dif$gl_flags [dif$v_parallel]               ! Call appropriate routine
;  195     0496  2     THEN output_parallel ()
;  196     0497  2
;  197     0498  2 ELSE IF .dif$gl_flags [dif$v_merged]
;  198     0499  2     THEN output_merged ()
;  199     0500  2
;  200     0501  2 ELSE IF .dif$gl_masfdb [fdb$v_separated]
;  201     0502  2     THEN output_separated (dif$gl_masfdb)
;  202     0503  2
;  203     0504  2 ELSE IF .dif$gl_revfdb [fdb$v_separated]
;  204     0505  2     THEN output_separated (dif$gl_revfdb)
;  205     0506  2
;  206     0507  2 ELSE IF .dif$gl_masfdb [fdb$v_changebar]
;  207     0508  2     THEN output_changebar (dif$gl_masfdb)
;  208     0509  2
;  209     0510  2 ELSE IF .dif$gl_revfdb [fdb$v_changebar]
;  210     0511  2     THEN output_changebar (dif$gl_revfdb)
;  211     0512  2
;  212     0513  2 ELSE IF .dif$gl_flags [dif$v_slp]
;  213     0514  2     THEN output_slp ();
;  214     0515  2
;  215     0516  2 RETURN true;
;  216     0517  1 END;


                                .TITLE  DIF_OUTPUT
                                .IDENT  \V04-000\

                                .PSECT  $OWN$,NOEXE,2
```

```
                                                00000 CMD_BUFPOS:
                                                      .BLKB   4
                                        04      00004 CR:       .BYTE   4
                                3E 52 43 3C     00005           .ASCII  \<CR>\                                          ;
                                                00009           .BLKB   3
                                        04      0000C FF:       .BYTE   4
                                3E 46 46 3C     0000D           .ASCII  \<FF>\                                          ;
                                                00011           .BLKB   3
                                        04      00014 LF:       .BYTE   4
                                3E 46 4C 3C     00015           .ASCII  \<LF>\                                          ;
                                                00019           .BLKB   3
                                        04      0001C VT:       .BYTE   4
                                3E 54 56 3C     0001D           .ASCII  \<VT>\                                          ;
                                                00021           .BLKB   3
                                        08      00024 BLANKS:   .BYTE   8
                        20 20 20 20 20 20 20 20 20      00025           .ASCII  \        \                              ;
                                                0002D           .BLKB   3
                                        01      00030 PERIOD:   .BYTE   1
                                        2E      00031           .ASCII  \.\                                             ;
                                                00032           .BLKB   2
                                        0D      00034 CHANGE:   .BYTE   13
        2A 2A 2A 45 47 4E 41 48 43 2A 2A 2A 20  00035           .ASCII  \ ***CHANGE***\                                 ;
                                                00042           .BLKB   2
                                        27      00044 DIFREC:   .BYTE   39
65 66 66 69 64 20 66 6F 20 72 65 62 6D 75 4E    00045           .ASCII  \Number of difference records found: !ZL\       ;
66 20 73 64 72 6F 63 65 72 20 65 63 6E 65 72    00054                                                                   ;
                  4C 5A 21 20 3A 64 6E 75 6F    00063                                                                   ;
                                        28      0006C DIFSEC:   .BYTE   40
65 66 66 69 64 20 66 6F 20 72 65 62 6D 75 4E    0006D           .ASCII  \Number of difference sections found: !ZL\      ;
20 73 6E 6F 69 74 63 65 73 20 65 63 6E 65 72    0007C                                                                   ;
                  4C 5A 21 20 3A 64 6E 75 6F 66 0008B                                                                   ;
                                                00095           .BLKB   3
                                        05      00098 FILE:     .BYTE   5
                                20 65 6C 69 46  00099           .ASCII  \File \                                         ;
                                                0009E           .BLKB   2
                                        18      000A0 HEXFULL:  .BYTE   24
5A 21 21 21 20 29 4C 58 39 28 4C 5A 21 21 21    000A1           .ASCII  \!!!ZL(9XL) !!!ZLAF !!6XL\                      ;
                  4C 58 36 21 21 20 46 41 4C    000B0                                                                   ;
                                                000B9           .BLKB   3
                                        2E      000BC HEXHEADER:
                                                        .BYTE   46                                                      ;
21 20 52 45 42 4D 55 4E 20 44 52 4F 43 45 52    000BD           .ASCII  \RECORD NUMBER !ZL (!8XL) LENGTH !ZL (!8X\      ;
54 47 4E 45 4C 20 29 4C 58 38 21 28 20 4C 5A    000CC                                                                   ;
                  58 38 21 28 20 4C 5A 21 20 48 000DB                                                                   ;
                           53 41 21 20 29 4C    000E5           .ASCII  \L) !AS\                                        ;
                                                000EB           .BLKB   1
                                        1F      000EC HEXPART:  .BYTE   31
21 20 29 4C 58 39 28 4C 5A 21 21 21 21 21 21    000ED           .ASCII  \!!!!!!ZL(9XL) !!!!!ZLAF !!!!6XL\               ;
58 36 21 21 21 21 20 46 41 4C 5A 21 21 21 21    000FC                                                                   ;
                                        4C      0010B                                                                   ;
                                        19      0010C OCTFULL:  .BYTE   25
21 21 21 20 29 4C 4F 32 31 28 4C 5A 21 21 21    0010D           .ASCII  \!!!ZL(12OL) !!!ZLAF !!6OL\                     ;
                  4C 4F 36 21 21 20 46 41 4C 5A 0011C                                                                   ;
                                                00126           .BLKB   2
                                        2E      00128 OCTHEADER:
                                                        .BYTE   46                                                      ;
```

```
21 20 52 45 42 4D 55 4E 20 44 52 4F 43 45 52  00129          .ASCII  \RECORD NUMBER !ZL (!8OL) LENGTH !ZL (!8O\  ;
54 47 4E 45 4C 20 29 4C 4F 38 21 28 20 4C 5A  00138                                                              ;
                  4F 38 21 28 20 4C 5A 21 20  00147                                                              ;
                              53 41 21 20 29  00151          .ASCII  \L) !AS\                                    ;
                                              00157          .BLKB   1                                           ;
                                           20 00158 OCTPART: .BYTE   32                                          ;
20 29 4C 4F 32 31 28 4C 5A 21 21 21 21 21 21  00159          .ASCII  \!!!!!!ZL(12OL) !!!!!ZLAF !!!!6OL\          ;
36 21 21 21 21 20 46 41 4C 5A 21 21 21 21 21  00160                                                              ;
                                       4C 4F  00177                                                              ;
                                              00179          .BLKB   3                                           ;
                                           06 0017C SLPOPRS: .BYTE   6                                           ;
                              3C 2F 40 25 5C 2D 0017D          .ASCII  \-\<92>\%a/<\                               ;
                                              00183          .BLKB   1                                           ;
                                           0C 00184 STARS:   .BYTE   12                                          ;
      2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A  00185          .ASCII  \************\                                ;

                                                          .EXTRN  DIF$GL_COMMDESC
                                                          .EXTRN  DIF$GL_CMDESC, DIF$GL_IGNORE
                                                          .EXTRN  DIF$GL_HEADER, DIF$GL_MATCH
                                                          .EXTRN  DIF$GL_MAXDIF, DIF$GL_MERGED
                                                          .EXTRN  DIF$GL_PARALLEL
                                                          .EXTRN  DIF$GL_WNDWSIZ, DIF$GL_FLAGS
                                                          .EXTRN  DIF$GL_DIFREC, DIF$GL_DIFSEC
                                                          .EXTRN  DIF$GL_DUMPWIDTH
                                                          .EXTRN  DIF$GL_ENTSPERLINE
                                                          .EXTRN  DIF$GL_WIDTH, DIF$GL_PARWIDTH
                                                          .EXTRN  DIF$GL_INBUF, DIF$GL_OUTBUF
                                                          .EXTRN  DIF$GL_FAOFULLBUF
                                                          .EXTRN  DIF$GL_FAOPARTBUF
                                                          .EXTRN  DIF$GL_FAOFULLDESC
                                                          .EXTRN  DIF$GL_FAOPARTDESC
                                                          .EXTRN  DIF$GL_MASDESC, DIF$GL_MASFDB
                                                          .EXTRN  DIF$GL_MASRAB, DIF$GL_REVDESC
                                                          .EXTRN  DIF$GL_REVFDB, DIF$GL_REVRAB
                                                          .EXTRN  DIF$GL_OUTDESC, DIF$GL_OUTRAB
                                                          .EXTRN  DIF$FORMAT_HEX_OCTAL
                                                          .EXTRN  OTS$CVT_L_TI, SYS$FAO
                                                          .EXTRN  DIF$_READERR, DIF$_WRITEERR

                                                          .PSECT  $CODE$,NOWRT,2

                                              001C 00000          .ENTRY  WRITE_MISMATCH, Save R2,R3,R4           ; 0467
                  54 00000000G   00    9E 00002          MOVAB   DIF$GL_FLAGS, R4                                ;
                  53 00000000G   00    9E 00009          MOVAB   DIF$GL_REVFDB+36, R3                            ;
                  52 00000000G   00    9E 00010          MOVAB   DIF$GL_MASFDB+36, R2                            ;
                     00000000G   00    D6 00017          INCL    DIF$GL_DIFSEC                                   ; 0493
09                64           06    E1 0001D          BBC     #6, DIF$GL_FLAGS, 1$                            ; 0495
   00000000V      EF           00    FB 00021          CALLS   #0, OUTPUT_PARALLEL                             ; 0496
                               48    11 00028          BRB     9$                                              ;
09                64           05    E1 0002A 1$:      BBC     #5, DIF$GL_FLAGS, 2$                            ; 0498
   00000000V      EF           00    FB 0002E          CALLS   #0, OUTPUT_MERGED                               ; 0499
                               3B    11 00035          BRB     9$                                              ;
05                62           02    E1 00037 2$:      BBC     #2, DIF$GL_MASFDB+36, 3$                        ; 0501
                      DC       A2    9F 0003B          PUSHAB  DIF$GL_MASFDB                                   ; 0502
                               07    11 0003E          BRB     4$                                              ;
0C                63           02    E1 00040 3$:      BBC     #2, DIF$GL_REVFDB+36, 5$                        ; 0504
                      DC       A3    9F 00044          PUSHAB  DIF$GL_REVFDB                                   ; 0505
```

```
                00000000V  EF              01  FB 00047 4$:    CALLS   #1, OUTPUT_SEPARATED                          ;
                                           22  11 0004E        BRB     9$                                            ;
                           05              62  E9 00050 5$:    BLBC    DIF$GL_MASFDB+36, 6$                          ; 0507
                                       DC  A2  9F 00053        PUSHAB  DIF$GL_MASFDB                                 ; 0508
                                           06  11 00056        BRB     7$                                            ;
                           0C              63  E9 00058 6$:    BLBC    DIF$GL_REVFDB+36, 8$                          ; 0510
                                       DC  A3  9F 0005B        PUSHAB  DIF$GL_REVFDB                                 ; 0511
                00000000V  EF              01  FB 0005E 7$:    CALLS   #1, OUTPUT_CHANGEBAR                          ;
                                           0B  11 00065        BRB     9$                                            ;
                           07          01  A4  E9 00067 8$:    BLBC    DIF$GL_FLAGS+1, 9$                            ; 0513
                00000000V  EF              00  FB 0006B        CALLS   #0, OUTPUT_SLP                                ; 0514
                           50              01  D0 00072 9$:    MOVL    #1, R0                                        ; 0516
                                           04  00075        RET                                                   ; 0517

; Routine Size:  118 bytes,    Routine Base:  $CODE$ + 0000
```

```
;   218          0518  1 GLOBAL ROUTINE additional_output =
;   219          0519  2 BEGIN
;   220          0520  2
;   221          0521  2 !++
;   222          0522  2 !
;   223          0523  2 ! FUNCTIONAL DESCRIPTION:
;   224          0524  2 !
;   225          0525  2 !       Output any additional listings, i.e. additional formats or radices,
;   226          0526  2 !       that have been requested.
;   227          0527  2 !
;   228          0528  2 ! INPUTS:
;   229          0529  2 !
;   230          0530  2 !       None
;   231          0531  2 !
;   232          0532  2 ! OUTPUTS:
;   233          0533  2 !
;   234          0534  2 !       None
;   235          0535  2 !
;   236          0536  2 ! ROUTINE VALUES:
;   237          0537  2 !
;   238          0538  2 !       Always true
;   239          0539  2 !
;   240          0540  2 !--
;   241          0541  2
;   242          0542  2 LOCAL
;   243          0543  2     first,                                  ! Flag set if finishing first listing
;   244          0544  2     save_flags : BBLOCK [4],                ! save dif$gl_flags
;   245          0545  2     dashdesc : BBLOCK [dsc$c_s_bln],        ! Descriptor for dashes
;   246          0546  2     stardesc : BBLOCK [dsc$c_s_bln];        ! Descriptor for stars
;   247          0547  2
;   248          0548  2 !
;   249          0549  2 ! If SLP output, then cannot have any more output forms, so return.
;   250          0550  2 !
;   251          0551  2 IF .dif$gl_flags [dif$v_slp]
;   252          0552  2     THEN RETURN true;
;   253          0553  2
;   254          0554  2 !
;   255          0555  2 ! Set up descriptor for stars.
;   256          0556  2 !
;   257          0557  2 stardesc [dsc$w_length] = .stars [0];
;   258          0558  2 stardesc [dsc$a_pointer] = stars [1];
;   259          0559  2
;   260          0560  2 !
;   261          0561  2 ! If init flag is still set, then no differences were found.
;   262          0562  2 ! So, cleanup current listing, but don't output any additional listings.
;   263          0563  2 !
;   264          0564  2 IF .dif$gl_flags [dif$v_init]
;   265          0565  3     THEN BEGIN
;   266          0566  3          IF NOT .dif$gl_flags [dif$v_parallel]
;   267          0567  3             AND NOT .dif$gl_flags [dif$v_merged]
;   268          0568  3             AND NOT .dif$gl_masfdb [fdb$v_separated]
;   269          0569  3             AND NOT .dif$gl_revfdb [fdb$v_separated]
;   270          0570  4             THEN BEGIN
;   271          0571  4                  put_desc (stardesc);
;   272          0572  4                  put_blank ();
;   273          0573  3                  END;
;   274          0574  3          RETURN true;
```

```
;  275          0575  2          END;
;  276          0576  2
;  277          0577  2      !
;  278          0578  2      ! If parallel output, then do special processing.
;  279          0579  2      !
;  280          0580  2      IF .dif$gl_flags [dif$v_parallel]
;  281          0581  2         THEN BEGIN
;  282          0582  3              first = false;                              ! Note that first listing is now done
;  283          0583  3              dashdesc [dsc$w_length] = .dif$gl_parwidth;     ! Build and output line of dashes
;  284          0584  3              dashdesc [dsc$a_pointer] = .dif$gl_outbuf;
;  285          0585  3              CH$FILL (%ASCII '-', .dif$gl_parwidth, .dif$gl_outbuf);
;  286          0586  3              put_desc (dashdesc);
;  287          0587  3              put_blank ();                               ! Output a blank line
;  288          0588  4              IF (NOT .dif$gl_masfdb [fdb$v_move] AND NOT .dif$gl_revfdb [fdb$v_move])
;  289          0589  4                 THEN RETURN true;                        ! No more listings? - then return
;  290          0590  3              END
;  291          0591  2         ELSE first = true;                              ! Note that first listing is not yet finished
;  292          0592  2
;  293          0593  2
;  294          0594  2      !
;  295          0595  2      ! If no more output, then use common (non-PARALLEL, non-SLP) ending,
;  296          0596  2      ! and return.
;  297          0597  2      !
;  298          0598  2      IF (NOT .dif$gl_masfdb [fdb$v_move] AND NOT .dif$gl_revfdb [fdb$v_move])
;  299          0599  2         THEN BEGIN
;  300          0600  3              IF NOT .dif$gl_flags [dif$v_merged]     ! Output stars for all but MERGED
;  301          0601  3                 THEN put_desc (stardesc);
;  302          0602  3              put_blank ();                           ! Output a blank line
;  303          0603  3              RETURN true;                            ! All done
;  304          0604  3              END;
;  305          0605  2
;  306          0606  2      !
;  307          0607  2      ! Save flags so they can be restored later
;  308          0608  2      !
;  309          0609  2      save_flags = .dif$gl_flags;
;  310          0610  2
;  311          0611  2      !
;  312          0612  2      ! Loop for all possible output formats.
;  313          0613  2      !
;  314          0614  2      INCR i FROM 1 TO 3                                  ! For each radix
;  315          0615  3      DO BEGIN
;  316          0616  3
;  317          0617  3      IF ((.i EQL 1) AND .dif$gl_flags [dif$v_ascii]) OR      ! If radix specified
;  318          0618  3         ((.i EQL 2) AND .dif$gl_flags [dif$v_hex]) OR
;  319          0619  4         ((.i EQL 3) AND .dif$gl_flags [dif$v_octal])
;  320          0620  4      THEN BEGIN                                          ! Then generate requested listings
;  321          0621  4
;  322          0622  5           IF (.i NEQ 1) AND (NOT .first)                 ! If hex or octal, and not already inited
;  323          0623  4              THEN init_hex_octal ();                     ! Then init
;  324          0624  4
;  325          0625  4      !
;  326          0626  4      ! For each listing format,
;  327          0627  4      !
;  328          0628  4      !       1. If first listing, then
;  329          0629  4      !                reset flag,
;  330          0630  4      !                output terminating lines.
;  331          0631  4      !
```

```
;  332     0632  4 !       2. If not first listing, then
;  333     0633  4 !             set init flag so that header will be output
;  334     0634  4 !             output listing,
;  335     0635  4 !             output terminating lines.
;  336     0636  4 !
;  337     0637  4
;  338     0638  4         IF .dif$gl_flags [dif$v_merged]             ! Generate MERGED output
;  339     0639  5         THEN BEGIN
;  340     0640  5              IF .first
;  341     0641  5              THEN first = false
;  342     0642  6              ELSE BEGIN
;  343     0643  6                   dif$gl_flags [dif$v_init] = true;
;  344     0644  6                   output_merged ();
;  345     0645  5                   END;
;  346     0646  5              put_blank ();
;  347     0647  4              END;
;  348     0648  4
;  349     0649  4         IF .dif$gl_masfdb [fdb$v_separated]         ! Generate SEPARATED output for master file
;  350     0650  5         THEN BEGIN
;  351     0651  5              IF .first
;  352     0652  5              THEN first = false
;  353     0653  6              ELSE BEGIN
;  354     0654  6                   dif$gl_flags [dif$v_init] = true;
;  355     0655  6                   output_separated (dif$gl_masfdb);
;  356     0656  5                   END;
;  357     0657  5              put_desc (stardesc);
;  358     0658  5              put_blank ();
;  359     0659  4              END;
;  360     0660  4
;  361     0661  4         IF .dif$gl_revfdb [fdb$v_separated]         ! Generate SEPARATED output for revision file
;  362     0662  5         THEN BEGIN
;  363     0663  5              IF .first
;  364     0664  5              THEN first = false
;  365     0665  6              ELSE BEGIN
;  366     0666  6                   dif$gl_flags [dif$v_init] = true;
;  367     0667  6                   output_separated (dif$gl_revfdb);
;  368     0668  5                   END;
;  369     0669  5              put_desc (stardesc);
;  370     0670  5              put_blank ();
;  371     0671  4              END;
;  372     0672  4
;  373     0673  4         IF .dif$gl_masfdb [fdb$v_changebar]         ! Generate CHANGE_BAR output for master file
;  374     0674  5         THEN BEGIN
;  375     0675  5              IF .first
;  376     0676  5              THEN first = false
;  377     0677  6              ELSE BEGIN
;  378     0678  6                   dif$gl_flags [dif$v_init] = true;
;  379     0679  6                   output_changebar (dif$gl_masfdb);
;  380     0680  5                   END;
;  381     0681  5              put_desc (stardesc);
;  382     0682  5              put_blank ();
;  383     0683  4              END;
;  384     0684  4
;  385     0685  4         IF .dif$gl_revfdb [fdb$v_changebar]         ! Generate CHANGE_BAR output for revision file
;  386     0686  5         THEN BEGIN
;  387     0687  5              IF .first
;  388     0688  5              THEN first = false
```

```
;  389          0689  6                 ELSE BEGIN
;  390          0690  6                      dif$gl_flags [dif$v_init] = true;
;  391          0691  6                      output_changebar (dif$gl_revfdb);
;  392          0692  5                      END;
;  393          0693  5                 put_desc (stardesc);
;  394          0694  5                 put_blank ();
;  395          0695  4                 END;
;  396          0696  4
;  397          0697  3             END;
;  398          0698  3
;  399          0699  3         !
;  400          0700  3         ! If done outputing a radix, then clear that bit.
;  401          0701  3         !
;  402          0702  3         IF .i EQL 1
;  403          0703  3            THEN dif$gl_flags [dif$v_ascii] = false
;  404          0704  3            ELSE IF .i EQL 2
;  405          0705  3                 THEN dif$gl_flags [dif$v_hex] = false
;  406          0706  3                 ELSE dif$gl_flags [dif$v_octal] = false;
;  407          0707  3
;  408          0708  2         END;
;  409          0709  2
;  410          0710  2         !
;  411          0711  2         ! restore flags
;  412          0712  2         !
;  413          0713  2         dif$gl_flags [dif$v_ascii] = .save_flags [dif$v_ascii];
;  414          0714  2         dif$gl_flags [dif$v_hex] = .save_flags [dif$v_hex];
;  415          0715  2         dif$gl_flags [dif$v_octal] = .save_flags [dif$v_octal];
;  416          0716  2
;  417          0717  2         RETURN true;
;  418          0718  1         END;


                                                0FFC 00000          .ENTRY  ADDITIONAL_OUTPUT, Save R2,R3,R4,R5,R6,R7,- ; 0518
                                                                            R8,R9,R10,R11
                                5B 00000000V    EF   9E 00002       MOVAB   PUT_DESC, R11                               ;
                                5A 00000000V    EF   9E 00009       MOVAB   PUT_BLANK, R10                              ;
                                59 00000000G    00   9E 00010       MOVAB   DIF$GL_REVFDB+36, R9                        ;
                                58 00000000G    00   9E 00017       MOVAB   DIF$GL_MASFDB+36, R8                        ;
                                57 00000000G    00   9E 0001E       MOVAB   DIF$GL_FLAGS, R7                            ;
                                5E                   10   C2 00025  SUBL2   #16, SP                                    ;
                                72                01 A7   E8 00028  BLBS    DIF$GL_FLAGS+1, 6$                        ; 0551
                                6E 00000000'         EF   9B 0002C  MOVZBW  STARS, STARDESC                           ; 0557
                             04 AE 00000000'         EF   9E 00033  MOVAB   STARS+1, STARDESC+4                       ; 0558
                          12 01 A7                   05   E1 00038  BBC     #5, DIF$GL_FLAGS+1, 1$                    ; 0564
                          5A    67                   06   E0 00040  BBS     #6, DIF$GL_FLAGS, 6$                      ; 0566
                          56    67                   05   E0 00044  BBS     #5, DIF$GL_FLAGS, 6$                      ; 0567
                          52    68                   02   E0 00048  BBS     #2, DIF$GL_MASFDB+36, 6$                  ; 0568
                          4E    69                   02   E0 0004C  BBS     #2, DIF$GL_REVFDB+36, 6$                  ; 0569
                                                     44   11 00050  BRB     4$                                        ; 0571
                          31    67                   06   E1 00052 1$:    BBC     #6, DIF$GL_FLAGS, 2$              ; 0580
                                                     56   D4 00056  CLRL    FIRST                                     ; 0582
                                51 00000000G         00   D0 00058  MOVL    DIF$GL_PARWIDTH, R1                       ; 0583
                             08 AE                   51   B0 0005F  MOVW    R1, DASHDESC                              ;
                                50 00000000G         00   D0 00063  MOVL    DIF$GL_OUTBUF, R0                         ; 0584
```

```
                        0C   AE          50   D0 0006A          MOVL    R0, DASHDESC+4
              51        2D   6E          00   2C 0006E          MOVC5   #0, (SP), #45, R1, (R0)                 ;0585
                             60          60       00073
                                     08  AE   9F 00074          PUSHAB  DASHDESC                                ;0586
                             6B          01   FB 00077          CALLS   #1, PUT_DESC
                             6A          00   FB 0007A          CALLS   #0, PUT_BLANK                           ;0587
                        20   68          03   E0 0007D          BBS     #3, DIF$GL_MASFDB+36, 7$               ;0588
                        05   69          03   E0 00081          BBS     #3, DIF$GL_REVFDB+36, 3$
                             56          17   11 00085          BRB     6$                                     ;0589
                             56          01   D0 00087 2$:      MOVL    #1, FIRST                              ;0591
                        13   68          03   E0 0008A 3$:      BBS     #3, DIF$GL_MASFDB+36, 7$               ;0598
                        0F   69          03   E0 0008E          BBS     #3, DIF$GL_REVFDB+36, 7$
                        05   67          05   E0 00092          BBS     #5, DIF$GL_FLAGS, 5$                   ;0600
                             5E          DD   00096 4$:         PUSHL   SP                                     ;0601
                             6B          01   FB 00098          CALLS   #1, PUT_DESC
                             6A          00   FB 0009B 5$:      CALLS   #0, PUT_BLANK                          ;0602
                           0105          31   0009E 6$:      BRW     33$                                    ;0603
                             54          67   D0 000A1 7$:      MOVL    DIF$GL_FLAGS, SAVE_FLAGS               ;0609
                             52          01   D0 000A4          MOVL    #1, I                                  ;0614
                                         53   D4 000A7 8$:      CLRL    R3                                     ;0617
                             01          52   D1 000A9          CMPL    I, #1
                                         05   12 000AC          BNEQ    9$
                                         53   D6 000AE          INCL    R3
                             15          67   E8 000B0          BLBS    DIF$GL_FLAGS, 13$
                             02          52   D1 000B3 9$:      CMPL    I, #2                                  ;0618
                                         04   12 000B6          BNEQ    10$
                        0C   67          01   E0 000B8          BBS     #1, DIF$GL_FLAGS, 13$
                             03          52   D1 000BC 10$:     CMPL    I, #3                                  ;0619
                                         03   13 000BF          BEQL    12$
                                       00AE   31 000C1 11$:     BRW     29$
                        F9   67          02   E1 000C4 12$:     BBC     #2, DIF$GL_FLAGS, 11$
                             01          52   D1 000C8 13$:     CMPL    I, #1                                  ;0622
                                         0A   13 000CB          BEQL    14$
                             07          56   E8 000CD          BLBS    FIRST, 14$
                 00000000V   EF          00   FB 000D0          CALLS   #0, INIT_HEX_OCTAL                     ;0623
              15             67          05   E1 000D7 14$:     BBC     #5, DIF$GL_FLAGS, 17$                  ;0638
                             04          56   E9 000DB          BLBC    FIRST, 15$                             ;0640
                                         56   D4 000DE          CLRL    FIRST                                  ;0641
                                         0B   11 000E0          BRB     16$
                          01 A7          20   88 000E2 15$:     BISB2   #32, DIF$GL_FLAGS+1                    ;0643
                 00000000V   EF          00   FB 000E6          CALLS   #0, OUTPUT_MERGED                      ;0644
                             6A          00   FB 000ED 16$:     CALLS   #0, PUT_BLANK                          ;0646
              1D             68          02   E1 000F0 17$:     BBC     #2, DIF$GL_MASFDB+36, 20$              ;0649
                             04          56   E9 000F4          BLBC    FIRST, 18$                             ;0651
                                         56   D4 000F7          CLRL    FIRST                                  ;0652
                                         0E   11 000F9          BRB     19$
                          01 A7          20   88 000FB 18$:     BISB2   #32, DIF$GL_FLAGS+1                    ;0654
                                   DC    A8   9F 000FF          PUSHAB  DIF$GL_MASFDB                          ;0655
                 00000000V   EF          01   FB 00102          CALLS   #1, OUTPUT_SEPARATED
                                         5E   DD 00109 19$:     PUSHL   SP                                     ;0657
                             6B          01   FB 0010B          CALLS   #1, PUT_DESC
                             6A          00   FB 0010E          CALLS   #0, PUT_BLANK                          ;0658
              1D             69          02   E1 00111 20$:     BBC     #2, DIF$GL_REVFDB+36, 23$              ;0661
                             04          56   E9 00115          BLBC    FIRST, 21$                             ;0663
                                         56   D4 00118          CLRL    FIRST                                  ;0664
                                         0E   11 0011A          BRB     22$
                          01 A7          20   88 0011C 21$:     BISB2   #32, DIF$GL_FLAGS+1                    ;0666
```

```
                                 DC  A9  9F  00120          PUSHAB  DIF$GL_REVFDB                                    ; 0667
                00000000V        EF  01  FB  00123          CALLS   #1, OUTPUT_SEPARATED
                                     5E  DD  0012A 22$:     PUSHL   SP                                              ; 0669
                                         6B  01  FB  0012C  CALLS   #1, PUT_DESC
                                         6A  00  FB  0012F  CALLS   #0, PUT_BLANK                                   ; 0670
                                         1D  68  E9  00132  BLBC    DIF$GL_MASFDB+36, 26$                           ; 0673
                                         04  56  E9  00135  BLBC    FIRST, 24$                                      ; 0675
                                             56  D4  00138  CLRL    FIRST                                           ; 0676
                                             0E  11  0013A  BRB     25$
                              01     A7      20  88  0013C 24$:    BISB2   #32, DIF$GL_FLAGS+1                     ; 0678
                                 DC  A8  9F  00140          PUSHAB  DIF$GL_MASFDB                                   ; 0679
                00000000V        EF  01  FB  00143          CALLS   #1, OUTPUT_CHANGEBAR
                                     5E  DD  0014A 25$:     PUSHL   SP                                              ; 0681
                                         6B  01  FB  0014C  CALLS   #1, PUT_DESC
                                         6A  00  FB  0014F  CALLS   #0, PUT_BLANK                                   ; 0682
                                         1D  69  E9  00152  BLBC    DIF$GL_REVFDB+36, 29$                           ; 0685
                                         04  56  E9  00155  BLBC    FIRST, 27$                                      ; 0687
                                             56  D4  00158  CLRL    FIRST                                           ; 0688
                                             0E  11  0015A  BRB     28$
                              01     A7      20  88  0015C 27$:    BISB2   #32, DIF$GL_FLAGS+1                     ; 0690
                                 DC  A9  9F  00160          PUSHAB  DIF$GL_REVFDB                                   ; 0691
                00000000V        EF  01  FB  00163          CALLS   #1, OUTPUT_CHANGEBAR
                                     5E  DD  0016A 28$:     PUSHL   SP                                              ; 0693
                                         6B  01  FB  0016C  CALLS   #1, PUT_DESC
                                         6A  00  FB  0016F  CALLS   #0, PUT_BLANK                                   ; 0694
                                         05  53  E9  00172  BLBC    R3, 30$                                         ; 0702
                                         67  01  8A  00175  BICB2   #1, DIF$GL_FLAGS                                ; 0703
                                             0D  11  00178  BRB     32$
                                         02      52  D1  0017A 30$:  CMPL    I, #2                                   ; 0704
                                             05  12  0017D  BNEQ    31$
                                         67      02  8A  0017F  BICB2   #2, DIF$GL_FLAGS                           ; 0705
                                             03  11  00182  BRB     32$
                                         67      04  8A  00184 31$:  BICB2   #4, DIF$GL_FLAGS                       ; 0706
                            FF1A     52      03  F1  00187 32$:  ACBL    #3, #1, I, 8$                           ; 0614
                              67     01      54  F0  0018D        INSV    SAVE_FLAGS, #0, #1, DIF$GL_FLAGS        ; 0713
                              50     54      01  EF  00192        EXTZV   #1, #1, SAVE_FLAGS, R0                  ; 0714
                              67     01      50  F0  00197        INSV    R0, #1, #1, DIF$GL_FLAGS
                              50     54      02  EF  0019C        EXTZV   #2, #1, SAVE_FLAGS, R0                  ; 0715
                              67     01      50  F0  001A1        INSV    R0, #2, #1, DIF$GL_FLAGS
                                             01  D0  001A6 33$:  MOVL    #1, R0                                  ; 0717
                                             04  001A9          RET                                             ; 0718

; Routine Size:  426 bytes,    Routine Base:  $CODE$ + 0076
```

```
;  420     0719  1 GLOBAL ROUTINE output_listing_trailer =
;  421     0720  2 BEGIN
;  422     0721  2
;  423     0722  2 !++
;  424     0723  2 !
;  425     0724  2 ! FUNCTIONAL DESCRIPTION:
;  426     0725  2 !
;  427     0726  2 !       Output the trailer for the listing.
;  428     0727  2 !
;  429     0728  2 ! INPUTS:
;  430     0729  2 !
;  431     0730  2 !       None.
;  432     0731  2 !
;  433     0732  2 ! OUTPUTS:
;  434     0733  2 !
;  435     0734  2 !       The trailer is output.
;  436     0735  2 !
;  437     0736  2 ! ROUTINE VALUES:
;  438     0737  2 !
;  439     0738  2 !       Always true
;  440     0739  2 !
;  441     0740  2 !--
;  442     0741  2 LOCAL
;  443     0742  2     linedesc : BBLOCK [dsc$c_s_bln],            ! Local string descs
;  444     0743  2     flag,
;  445     0744  2     mask,                                       ! mask
;  446     0745  2     outdesc : BBLOCK [dsc$c_s_bln];
;  447     0746  2
;  448     0747  2 cmd_bufpos = 0;                 ! initialize output routine
;  449     0748  2
;  450     0749  2 IF .dif$gl_flags [dif$v_slp]                    ! If SLP
;  451     0750  3    THEN BEGIN                                   ! Then just output slash
;  452     0751  3         linedesc [dsc$w_length] = 1;
;  453     0752  3         linedesc [dsc$a_pointer] = UPLIT('/');
;  454     0753  3         put_desc (linedesc);
;  455     0754  3         RETURN true;
;  456     0755  2         END;
;  457     0756  2
;  458     0757  2 !
;  459     0758  2 ! Output the number of difference sections.
;  460     0759  2 !
;  461     0760  2 output_cmdfao (difsec, .dif$gl_difsec);
;  462     0761  2 output_cmdentity (0);
;  463     0762  2
;  464     0763  2 !
;  465     0764  2 ! Output the number of difference records.
;  466     0765  2 !
;  467     0766  2 output_cmdfao (difrec, .dif$gl_difrec);
;  468     0767  2 output_cmdentity (0);
;  469     0768  2
;  470     0769  2 !
;  471     0770  2 ! Skip a line and then output the command line.
;  472     0771  2 !
;  473     0772  2
;  474     0773  2 put_blank ();
;  475     0774  2
;  476     0775  2 output_cmdcounted (cstring ('DIFFERENCES '));
```

```
;  477          0776  2
;  478          0777  2 IF .dif$gl_flags [dif$v_ignore]          ! /IGNORE
;  479          0778  2 THEN
;  480          0779  2     BEGIN
;  481          0780  2     BIND
;  482          0781  2         list = PLIT (
;  483          0782  2             UPLIT (ign$m_blnklin,       %ASCIC 'BLANK_LINES'),
;  484          0783  2             UPLIT (ign$m_comments,      %ASCIC 'COMMENTS'),
;  485          0784  2             UPLIT (ign$m_exact,         %ASCIC 'EXACT'),
;  486          0785  2             UPLIT (ign$m_formfeed,      %ASCIC 'FORM_FEEDS'),
;  487          0786  2             UPLIT (ign$m_header,        %ASCIC 'HEADER'),
;  488          0787  2             UPLIT (ign$m_pretty,        %ASCIC 'PRETTY'),
;  489          0788  2             UPLIT (ign$m_traiblnk,      %ASCIC 'TRAILING_SPACES'),
;  490          0789  2             UPLIT (ign$m_spacing,       %ASCIC 'SPACING'); : VECTOR [, LONG];
;  491          0790  2     mask = 0;
;  492          0791  2     DECR i FROM .list [-1] - 1 TO 0
;  493          0792  2     DO
;  494          0793  2         mask = .mask OR (..list [.i] AND .dif$gl_ignore);
;  495          0794  2     output_cmdcounted (cstring ('/IGNORE=('));
;  496          0795  2     DECR i FROM .list [-1] - 1 TO 0
;  497          0796  2     DO
;  498          0797  2         IF (..list [.i] AND .mask) NEQ 0
;  499          0798  2         THEN
;  500          0799  4             BEGIN
;  501          0800  4             mask = .mask AND NOT ..list [.i];
;  502          0801  4             output_cmdcounted (.list [.i]+4);
;  503          0802  4             IF (..list [.i] AND ign$m_header) NEQ 0     ! special case HEADER
;  504          0803  4             THEN
;  505          0804  4                 output_cmdfao (cstring ('=!UL'), .dif$gl_header);
;  506          0805  4             IF .mask NEQ 0
;  507          0806  4             THEN
;  508          0807  4                 output_cmdcounted (cstring (','));
;  509          0808  3             END;
;  510          0809  3     output_cmdcounted (cstring (')'));
;  511          0810  2     END;
;  512          0811  2
;  513          0812  2 IF .dif$gl_flags [dif$v_comdel]          ! /COMMENT_DELIMITERS
;  514          0813  2 THEN
;  515          0814  3     BEGIN
;  516          0815  3     output_cmdcounted (cstring ('/COMMENT_DELIMITERS'));
;  517          0816  3     IF .dif$gl_commdesc [dsc$w_length] NEQ 0
;  518          0817  3     THEN
;  519          0818  4         BEGIN
;  520          0819  4         output_cmdcounted (cstring ('=('));
;  521          0820  4         INCR i FROM 0 to .dif$gl_commdesc [dsc$w_length]-1
;  522          0821  4         DO
;  523          0822  5             BEGIN
;  524          0823  5             SELECTONE CH$RCHAR (.dif$gl_commdesc [dsc$a_pointer]+.i) OF
;  525          0824  5             SET
;  526          0825  5             [%C':'] : output_cmdcounted (cstring ('COLON'));
;  527          0826  5             [%C','] : output_cmdcounted (cstring ('COMMA'));
;  528          0827  5             [%C'!'] : output_cmdcounted (cstring ('EXCLAMATION'));
;  529          0828  5             [ffeed] : output_cmdcounted (cstring ('FORM_FEED'));
;  530          0829  5             [%C'['] : output_cmdcounted (cstring ('LEFT'));
;  531          0830  5             [%C']'] : output_cmdcounted (cstring ('RIGHT'));
;  532          0831  5             [%C';'] : output_cmdcounted (cstring ('SEMI_COLON'));
;  533          0832  5             [%C'/'] : output_cmdcounted (cstring ('SLASH'));
```

```
;  534     0833  5              [%C' '] : output_cmdcounted (cstring ('SPACE'));
;  535     0834  5              [%C'	'] : output_cmdcounted (cstring ('TAB'));
;  536     0835  5              [OTHERWISE] :
;  537     0836  6                  BEGIN
;  538     0837  6                  outdesc [dsc$w_length] = 1;
;  539     0838  6                  outdesc [dsc$a_pointer] = .dif$gl_commdesc [dsc$a_pointer]+.i;
;  540     0839  6                  output_cmdfao (cstring ('"!AS"'), outdesc);
;  541     0840  5                  END;
;  542     0841  5              TES;
;  543     0842  5              IF .i NEQ .dif$gl_commdesc [dsc$w_length]-1
;  544     0843  5              THEN
;  545     0844  5                  output_cmdcounted (cstring (','));
;  546     0845  4              END;
;  547     0846  4          output_cmdcounted (cstring (')'));
;  548     0847  3          END;
;  549     0848  2      END;
;  550     0849  2
;  551     0850  2  IF .dif$gl_flags [dif$v_width]                  ! /WIDTH or /LINE_WIDTH
;  552     0851  2  THEN
;  553     0852  2      output_cmdfao (cstring ('/WIDTH=!UL'), .dif$gl_width);
;  554     0853  2
;  555     0854  2  IF .dif$gl_flags [dif$v_match]                  ! /MATCH
;  556     0855  2  THEN
;  557     0856  2      output_cmdfao (cstring ('/MATCH=!UL'), .dif$gl_match);
;  558     0857  2
;  559     0858  2  IF .dif$gl_flags [dif$v_maxdif]                 ! /MAXIMUM_DIFFERENCES
;  560     0859  2  THEN
;  561     0860  2      output_cmdfao (cstring ('/MAXIMUM_DIFFERENCES=!UL'), .dif$gl_maxdif);
;  562     0861  2
;  563     0862  2  IF .dif$gl_flags [dif$v_merged]                 ! /MERGED
;  564     0863  2  THEN
;  565     0864  2      output_cmdfao (cstring ('/MERGED=!UL'), .dif$gl_merged);
;  566     0865  2
;  567     0866  2  IF NOT .dif$gl_flags [dif$v_ascii]              ! /MODE
;  568     0867  2      OR .dif$gl_flags [dif$v_hex]
;  569     0868  2      OR .dif$gl_flags [dif$v_octal]
;  570     0869  2  THEN
;  571     0870  3      BEGIN
;  572     0871  3      BIND
;  573     0872  3          list = PLIT (
;  574     0873  3              UPLIT (dif$m_ascii,          %ASCIC 'ASCII'),
;  575     0874  3              UPLIT (dif$m_hex,            %ASCIC 'HEXADECIMAL'),
;  576     0875  3              UPLIT (dif$m_octal,          %ASCIC 'OCTAL')) : VECTOR [, LONG];
;  577     0876  3      mask = 0;
;  578     0877  3      DECR i FROM .list [-1] - 1 TO 0
;  579     0878  3      DO
;  580     0879  3          mask = .mask OR (..list [.i] AND .dif$gl_flags);
;  581     0880  3      output_cmdcounted (cstring ('/MODE=('));
;  582     0881  3      DECR i FROM .list [-1] - 1 TO 0
;  583     0882  3      DO
;  584     0883  3          IF (..list [.i] AND .mask) NEQ 0
;  585     0884  3          THEN
;  586     0885  4              BEGIN
;  587     0886  4              mask = .mask AND NOT ..list [.i];
;  588     0887  4              output_cmdcounted (.list [.i]+4);
;  589     0888  4              IF .mask NEQ 0
;  590     0889  4              THEN
```

```
;  591     0890  4                output_cmdcounted (cstring (','));
;  592     0891  4            END;
;  593     0892  3        output_cmdcounted (cstring (')'));
;  594     0893  2        END;
;  595     0894  2
;  596     0895  2    IF .dif$gl_flags [dif$v_output]                  ! /OUTPUT
;  597     0896  2    THEN
;  598     0897  3        BEGIN
;  599     0898  3        output_cmdcounted (cstring ('/OUTPUT='));
;  600     0899  3        output_cmdentity (dif$gl_outdesc);
;  601     0900  2        END;
;  602     0901  2
;  603     0902  2    IF .dif$gl_flags [dif$v_parallel]                ! /PARALLEL
;  604     0903  2    THEN
;  605     0904  2        output_cmdcounted (cstring ('/PARALLEL'));
;  606     0905  2
;  607     0906  2    IF .dif$gl_masfdb [fdb$v_separated] AND .dif$gl_revfdb [fdb$v_separated]
;  608     0907  2    THEN
;  609     0908  2        output_cmdcounted (cstring ('/SEPARATED'))
;  610     0909  2    ELSE IF .dif$gl_masfdb [fdb$v_separated]
;  611     0910  2    THEN
;  612     0911  2        output_cmdcounted (cstring ('/SEPARATED=MASTER'))
;  613     0912  2    ELSE IF .dif$gl_revfdb [fdb$v_separated]
;  614     0913  2    THEN
;  615     0914  2        output_cmdcounted (cstring ('/SEPARATED=REVISION'));
;  616     0915  2
;  617     0916  2    IF .dif$gl_flags [dif$v_slp]                     ! /SLP
;  618     0917  2    THEN
;  619     0918  2        output_cmdcounted (cstring ('/SLP'));
;  620     0919  2
;  621     0920  2    IF .dif$gl_flags [dif$v_window]                  ! /WINDOW
;  622     0921  2    THEN
;  623     0922  2        output_cmdfao (cstring ('/WINDOW=!UL'), .dif$gl_wndwsiz);
;  624     0923  2
;  625     0924  2    IF NOT .dif$gl_flags [dif$v_linenum]             ! /NUMBER
;  626     0925  2    THEN
;  627     0926  2        output_cmdcounted (cstring ('/NONUMBER'));
;  628     0927  2
;  629     0928  2    output_cmdcounted (cstring (''));                ! terminate line
;  630     0929  2
;  631     0930  2    output_cmdfile (dif$gl_masfdb);                  ! output master file line
;  632     0931  2    output_cmdcounted (cstring (''));                ! terminate line
;  633     0932  2
;  634     0933  2    output_cmdfile (dif$gl_revfdb);                  ! output revision file line
;  635     0934  2    output_cmdentity (0);                            ! finish the output
;  636     0935  2
;  637     0936  2    ! put_desc (dif$gl_cmdesc);                      ! original command line
;  638     0937  2
;  639     0938  2    RETURN true;
;  640     0939  1    END;


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                00  00  00  2F  00000 P.AAA:  .ASCII  \/\<0><0><0>                    ;
                                            0C  00004 P.AAB:  .BYTE   12                              ;
```

```
                20 53 45 43 4E 45 52 45 46 46 49 44  00005          .ASCII  \DIFFERENCES \                                  ;
                                                     00011          .BLKB   3                                               ;
                                   00000001          00014 P.AAD:   .LONG   1                                               ;
                53 45 4E 49 4C 5F 4B 4E 41 4C 42 0B  00018          .ASCII  <11>\BLANK_LINES\                               ;
                                   00000002          00024 P.AAE:   .LONG   2                                               ;
                00 00 00 53 54 4E 45 4D 4D 4F 43 08  00028          .ASCII  <8>\COMMENTS\<0><0><0>                          ;
                                   00000040          00034 P.AAF:   .LONG   64                                              ;
                         00 00 54 43 41 58 45 05     00038          .ASCII  <5>\EXACT\<0><0>                                ;
                                   00000004          00040 P.AAG:   .LONG   4                                               ;
                00 53 44 45 45 46 5F 4D 52 4F 46 0A  00044          .ASCII  <10>\FORM_FEEDS\<0>                             ;
                                   00000020          00050 P.AAH:   .LONG   32                                              ;
                         00 52 45 44 41 45 48 06     00054          .ASCII  <6>\HEADER\<0>                                  ;
                                   00000080          0005C P.AAI:   .LONG   128                                             ;
                         00 59 54 54 45 52 50 06     00060          .ASCII  <6>\PRETTY\<0>                                  ;
                                   00000008          00068 P.AAJ:   .LONG   8                                               ;
45 43 41 50 53 5F 47 4E 49 4C 49 41 52 54 0F         0006C          .ASCII  <15>\TRAILING_SPACES\                           ;
                                                     0007B                                                                  ;
                                   00000010          0007C P.AAK:   .LONG   16                                              ;
                         47 4E 49 43 41 50 53 07     00080          .ASCII  <7>\SPACING\                                    ;
                                   00000008          00088          .LONG   8                                               ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0008C P.AAC:   .ADDRESS P.AAD, P.AAE, P.AAF, P.AAG, P.AAH, -   ;
                                   00000000' 00000000' 000A4                P.AAI, P.AAJ, P.AAK                           ;
                                                     09  000AC P.AAL:   .BYTE   9                                           ;
                28 3D 45 52 4F 4E 47 49 2F           000AD          .ASCII  \/IGNORE=(\                                     ;
                                                     04  000B6 P.AAM:   .BYTE   4                                           ;
                            4C 55 21 3D              000B7          .ASCII  \=!UL\                                          ;
                                                     01  000BB P.AAN:   .BYTE   1                                           ;
                                                     2C  000BC          .ASCII  \,\                                         ;
                                                     01  000BD P.AAO:   .BYTE   1                                           ;
                                                     29  000BE          .ASCII  \)\                                         ;
                                                     13  000BF P.AAP:   .BYTE   19                                          ;
49 4D 49 4C 45 44 5F 54 4E 45 4D 4D 4F 43 2F         000C0          .ASCII  \/COMMENT_DELIMITERS\                           ;
                                  53 52 45 54        000CF                                                                  ;
                                                     02  000D3 P.AAQ:   .BYTE   2                                           ;
                                        28 3D        000D4          .ASCII  \=(\                                            ;
                                                     05  000D6 P.AAR:   .BYTE   5                                           ;
                               4E 4F 4C 4F 43        000D7          .ASCII  \COLON\                                         ;
                                                     05  000DC P.AAS:   .BYTE   5                                           ;
                               41 4D 4D 4F 43        000DD          .ASCII  \COMMA\                                         ;
                                                     0B  000E2 P.AAT:   .BYTE   11                                          ;
                4E 4F 49 54 41 4D 41 4C 43 58 45     000E3          .ASCII  \EXCLAMATION\                                   ;
                                                     09  000EE P.AAU:   .BYTE   9                                           ;
                   44 45 45 46 5F 4D 52 4F 46        000EF          .ASCII  \FORM_FEED\                                     ;
                                                     04  000F8 P.AAV:   .BYTE   4                                           ;
                                  54 46 45 4C        000F9          .ASCII  \LEFT\                                          ;
                                                     05  000FD P.AAW:   .BYTE   5                                           ;
                               54 48 47 49 52        000FE          .ASCII  \RIGHT\                                         ;
                                                     0A  00103 P.AAX:   .BYTE   10                                          ;
                   4E 4F 4C 4F 43 5F 49 4D 45 53     00104          .ASCII  \SEMI_COLON\                                    ;
                                                     05  0010E P.AAY:   .BYTE   5                                           ;
                               48 53 41 4C 53        0010F          .ASCII  \SLASH\                                         ;
                                                     05  00114 P.AAZ:   .BYTE   5                                           ;
                               45 43 41 50 53        00115          .ASCII  \SPACE\                                         ;
                                                     03  0011A P.ABA:   .BYTE   3                                           ;
                                     42 41 54        0011B          .ASCII  \TAB\                                           ;
                                                     05  0011E P.ABB:   .BYTE   5                                           ;
                               22 53 41 21 22        0011F          .ASCII  \"!AS"\                                         ;
```

```
                                                                                01  00124 P.ABC:  .BYTE   1                                        ;
                                                                                2C  00125      .ASCII  \,\                                      ;
                                                                                01  00126 P.ABD:  .BYTE   1                                        ;
                                                                                29  00127      .ASCII  \)\                                      ;
                                                                                0A  00128 P.ABE:  .BYTE   10                                       ;
                                  4C  55  21  3D  48  54  44  49  57            2F  00129      .ASCII  \/WIDTH=!UL\                             ;
                                                                                0A  00133 P.ABF:  .BYTE   10                                       ;
                                  4C  55  21  3D  48  43  54  41  4D            2F  00134      .ASCII  \/MATCH=!UL\                             ;
                                                                                18  0013E P.ABG:  .BYTE   24                                       ;
52  45  46  46  49  44  5F  4D  55  4D  49  58  41  4D                           2F  0013F      .ASCII  \/MAXIMUM_DIFFERENCES=!UL\               ;
                          4C  55  21  3D  53  45  43  4E  45                        0014E                                                       ;
                                                                                0B  00157 P.ABH:  .BYTE   11                                       ;
                      4C  55  21  3D  44  45  47  52  45  4D                    2F  00158      .ASCII  \/MERGED=!UL\                            ;
                                                                                    00163      .BLKB   1                                       ;
                                                                  00000001          00164 P.ABJ:  .LONG   1                                        ;
                                  00  00  49  49  43  53  41  05                    00168      .ASCII  <5>\ASCII\<0><0>                        ;
                                                                  00000002          00170 P.ABK:  .LONG   2                                        ;
                  4C  41  4D  49  43  45  44  41  58  45  48  0B                    00174      .ASCII  <11>\HEXADECIMAL\                       ;
                                                                  00000004          00180 P.ABL:  .LONG   4                                        ;
                                  00  00  4C  41  54  43  4F  05                    00184      .ASCII  <5>\OCTAL\<0><0>                        ;
                                                                  00000003          0018C      .LONG   3                                       ;
                              00000000'  00000000'  00000000'                       00190 P.ABI:  .ADDRESS P.ABJ, P.ABK, P.ABL                 ;
                                                                                07  0019C P.ABM:  .BYTE   7                                        ;
                                      28  3D  45  44  4F  4D  2F                    0019D      .ASCII  \/MODE=(\                               ;
                                                                                01  001A4 P.ABN:  .BYTE   1                                        ;
                                                                                2C  001A5      .ASCII  \,\                                      ;
                                                                                01  001A6 P.ABO:  .BYTE   1                                        ;
                                                                                29  001A7      .ASCII  \)\                                      ;
                                                                                08  001A8 P.ABP:  .BYTE   8                                        ;
                                  3D  54  55  50  54  55  4F                    2F  001A9      .ASCII  \/OUTPUT=\                               ;
                                                                                09  001B1 P.ABQ:  .BYTE   9                                        ;
                              4C  45  4C  4C  41  52  41  50                    2F  001B2      .ASCII  \/PARALLEL\                              ;
                                                                                0A  001BB P.ABR:  .BYTE   10                                       ;
                          44  45  54  41  52  41  50  45  53                    2F  001BC      .ASCII  \/SEPARATED\                             ;
                                                                                11  001C6 P.ABS:  .BYTE   17                                       ;
54  53  41  4D  3D  44  45  54  41  52  41  50  45  53                           2F  001C7      .ASCII  \/SEPARATED=MASTER\                      ;
                                                      52                        45  001D6                                                       ;
                                                                                13  001D8 P.ABT:  .BYTE   19                                       ;
49  56  45  52  3D  44  45  54  41  52  41  50  45  53                           2F  001D9      .ASCII  \/SEPARATED=REVISION\                    ;
                                                  4E  4F  49                        001E8                                                       ;
                                                                                04  001EC P.ABU:  .BYTE   4                                        ;
                                                  50  4C  53                    2F  001ED      .ASCII  \/SLP\                                   ;
                                                                                0B  001F1 P.ABV:  .BYTE   11                                       ;
                      4C  55  21  3D  57  4F  44  4E  49  57                    2F  001F2      .ASCII  \/WINDOW=!UL\                            ;
                                                                                09  001FD P.ABW:  .BYTE   9                                        ;
                              52  45  42  4D  55  4E  4F  4E                    2F  001FE      .ASCII  \/NONUMBER\                              ;
                                                                                00  00207 P.ABX:  .BYTE   0                                        ;
                                                                                    00208      .BLKB   0                                       ;
                                                                                00  00208 P.ABY:  .BYTE   0                                        ;
                                                                                    00209      .BLKB   0                                       ;

                                                                                    LIST=              P.AAC
                                                                                    LIST=              P.ABI


                                                                                        .PSECT  $CODE$,NOWRT,2
```

```
                                 0FFC 00000          .ENTRY  OUTPUT_LISTING_TRAILER, Save R2,R3,R4,R5,-  ; 0719
                                                             R6,R7,R8,R9,R10,R11
                          5B 00000000V EF  9E 00002          MOVAB   OUTPUT_CMDENTITY, R11                ;
                          5A 00000000V EF  9E 00009          MOVAB   OUTPUT_CMDFAO, R10                   ;
                          59 00000000G 00  9E 00010          MOVAB   DIF$GL_FLAGS, R9                     ;
                          58 00000000V EF  9E 00017          MOVAB   OUTPUT_CMDCOUNTED, R8                ;
                          57 00000000' EF  9E 0001E          MOVAB   LIST, R7                             ;
                          5E           10  C2 00025          SUBL2   #16, SP                              ;
                             00000000' EF  D4 00028          CLRL    CMD_BUFPOS                           ; 0747
                          17    01     A9  E9 0002E          BLBC    DIF$GL_FLAGS+1, 1$                   ; 0749
                       08 AE            01  B0 00032          MOVW    #1, LINEDESC                         ; 0751
                       0C AE      FF74  C7  9E 00036          MOVAB   P.AAA, LINEDESC+4                    ; 0752
                             08 AE      9F  9E 0003C          PUSHAB  LINEDESC                             ; 0753
                0000000CV EF            01  FB 0003F          CALLS   #1, PUT_DESC                         ;
                                  02BE  31  00046          BRW     46$                                  ; 0754
                          00000000G 00  DD  00049 1$:       PUSHL   DIF$GL_DIFSEC                        ; 0760
                          00000000' EF  9F  0004F          PUSHAB  DIFSEC                               ;
                          6A            02  FB 00055          CALLS   #2, OUTPUT_CMDFAO                    ;
                                        7E  D4 00058          CLRL    -(SP)                                ; 0761
                          6B            01  FB 0005A          CALLS   #1, OUTPUT_CMDENTITY                 ;
                          00000000G 00  DD  0005D          PUSHL   DIF$GL_DIFREC                        ; 0766
                          00000000' EF  9F  00063          PUSHAB  DIFREC                               ;
                          6A            02  FB 00069          CALLS   #2, OUTPUT_CMDFAO                    ;
                                        7E  D4 0006C          CLRL    -(SP)                                ; 0767
                          6B            01  FB 0006E          CALLS   #1, OUTPUT_CMDENTITY                 ;
                00000000V EF            00  FB 00071          CALLS   #0, PUT_BLANK                        ; 0773
                             FF78       C7  9F 00078          PUSHAB  P.AAB                                ; 0775
                          68            01  FB 0007C          CALLS   #1, OUTPUT_CMDCOUNTED                ;
                       62               04  E1 0007F          BBC     #4, DIF$GL_FLAGS, 7$                 ; 0777
                                        55  D4 00083          CLRL    MASK                                 ; 0790
                          50      FC    A7  D0 00085          MOVL    LIST-4, I                            ; 0791
                                        12  11 00089          BRB     3$                                   ;
                          51          6740  D0 0008B 2$:      MOVL    LIST[I], R1                          ; 0793
                          52 00000000G  00  D2 0008F          MCOML   DIF$GL_IGNORE, R2                    ;
                       51 61            52  CB 00096          BICL3   R2, (R1), R1                         ;
                          55            51  C8 0009A          BISL2   R1, MASK                             ;
                          EB            50  F4 0009D 3$:      SOBGEQ  I, 2$                                ;
                                20      A7  9F 000A0          PUSHAB  P.AAL                                ; 0794
                          68            01  FB 000A3          CALLS   #1, OUTPUT_CMDCOUNTED                ;
                          52      FC    A7  D0 000A6          MOVL    LIST-4, I                            ; 0795
                                        30  11 000AA          BRB     6$                                   ;
                          50          6742  D0 000AC 4$:      MOVL    LIST[I], R0                          ; 0797
                          55            60  D3 000B0          BITL    (R0), MASK                           ;
                                        27  13 000B3          BEQL    6$                                   ;
                          55            60  CA 000B5          BICL2   (R0), MASK                           ; 0800
                                04      A0  9F 000B8          PUSHAB  4(R0)                                ; 0801
                          68            01  FB 000BB          CALLS   #1, OUTPUT_CMDCOUNTED                ;
                          50          6742  D0 000BE          MOVL    LIST[I], R0                          ; 0802
                       0C 60            05  E1 000C2          BBC     #5, (R0), 5$                         ;
                          00000000G 00  DD  000C6          PUSHL   DIF$GL_HEADER                        ; 0804
                                2A      A7  9F 000CC          PUSHAB  P.AAM                                ;
                          6A            02  FB 000CF          CALLS   #2, OUTPUT_CMDFAO                    ;
                                        55  D5 000D2 5$:      TSTL    MASK                                 ; 0805
                                        06  13 000D4          BEQL    6$                                   ;
                                2F      A7  9F 000D6          PUSHAB  P.AAN                                ; 0807
                          68            01  FB 000D9          CALLS   #1, OUTPUT_CMDCOUNTED                ;
                          CD            52  F4 000DC 6$:      SOBGEQ  I, 4$                                ; 0797
```

```
                        31   A7  9F 000DF          PUSHAB  P.AAO                                   ; 0809
                    68       01  FB 000E2          CALLS   #1, OUTPUT_CMDCOUNTED
            03      69       03  E0 000E5 7$:      BBS     #3, DIF$GL_FLAGS, 9$                    ; 0812
                      00CE   31  000E9 8$:          BRW     26$
                        33   A7  9F 000EC 9$:      PUSHAB  P.AAP                                   ; 0815
                    68       01  FB 000EF          CALLS   #1, OUTPUT_CMDCOUNTED
                00000000G    00  B5 000F2          TSTW    DIF$GL_COMMDESC                         ; 0816
                             EF  13 000F8          BEQL    8$
                        47   A7  9F 000FA          PUSHAB  P.AAQ                                   ; 0819
                    68       01  FB 000FD          CALLS   #1, OUTPUT_CMDCOUNTED
                56 00000000G 00  3C 00100          MOVZWL  DIF$GL_COMMDESC, R6                     ; 0820
                54           01  CE 00107          MNEGL   #1, I
                      009D   31  00110            BRW     23$
                53 00000000G 00  D0 00113 10$:     MOVL    DIF$GL_COMMDESC+4, R3                   ; 0823
                52         6443  9A 00117          MOVZBL  (I)[R3], R2
                3A           52  91 0011B          CMPB    R2, #58                                 ; 0825
                             05  12 0011E          BNEQ    11$
                        4A   A7  9F 00120          PUSHAB  P.AAR
                             5D  11 00123          BRB     20$
                2C           52  91 00125 11$:     CMPB    R2, #44                                 ; 0826
                             05  12 00128          BNEQ    12$
                        50   A7  9F 0012A          PUSHAB  P.AAS
                             53  11 0012D          BRB     20$
                21           52  91 0012F 12$:     CMPB    R2, #33                                 ; 0827
                             05  12 00132          BNEQ    13$
                        56   A7  9F 00134          PUSHAB  P.AAT
                             49  11 00137          BRB     20$
                0C           52  91 00139 13$:     CMPB    R2, #12                                 ; 0828
                             05  12 0013C          BNEQ    14$
                        62   A7  9F 0013E          PUSHAB  P.AAU
                             3F  11 00141          BRB     20$
            5B  8F           52  91 00143 14$:     CMPB    R2, #91                                 ; 0829
                             05  12 00147          BNEQ    15$
                        6C   A7  9F 00149          PUSHAB  P.AAV
                             34  11 0014C          BRB     20$
            5D  8F           52  91 0014E 15$:     CMPB    R2, #93                                 ; 0830
                             05  12 00152          BNEQ    16$
                        71   A7  9F 00154          PUSHAB  P.AAW
                             29  11 00157          BRB     20$
                3A           52  91 00159 16$:     CMPB    R2, #58                                 ; 0831
                             05  12 0015C          BNEQ    17$
                        77   A7  9F 0015E          PUSHAB  P.AAX
                             1F  11 00161          BRB     20$
                2F           52  91 00163 17$:     CMPB    R2, #47                                 ; 0832
                             06  12 00166          BNEQ    18$
                      0082   C7  9F 00168          PUSHAB  P.AAY
                             14  11 0016C          BRB     20$
                20           52  91 0016E 18$:     CMPB    R2, #32                                 ; 0833
                             06  12 00171          BNEQ    19$
                      008C   C7  9F 00173          PUSHAB  P.AAZ
                             09  11 00177          BRB     20$
                09           52  91 00179 19$:     CMPB    R2, #9                                  ; 0834
                             09  12 0017C          BNEQ    21$
                      008E   C7  9F 0017E          PUSHAB  P.ABA
                68           01  FB 00182 20$:     CALLS   #1, OUTPUT_CMDCOUNTED
                             11  11 00185          BRB     22$
                6E           01  B0 00187 21$:     MOVW    #1, OUTDESC                             ; 0837
```

```
                04 AE          53              54 C1 00187          ADDL3   I, R3, OUTDESC+4                        ; 0838
                                               5E DD 0018C          PUSHL   SP                                      ; 0839
                                     0092      C7 9F 0018E          PUSHAB  P.ABB
                                            6A 02 FB 00192          CALLS   #2, OUTPUT_CMDFAO
                                            50 00000000G   00 3C 00195 22$:  MOVZWL  DIF$GL_COMMDESC, R0            ; 0842
                                               50 D7 0019C          DECL    R0
                                            50             54 D1 0019E          CMPL    I, R0
                                               07 13 001A1          BEQL    23$
                                     0098      C7 9F 001A3          PUSHAB  P.ABC                                   ; 0844
                                            68 01 FB 001A7          CALLS   #1, OUTPUT_CMDCOUNTED
                                     02        54             56 F2 001AA 23$:  AOBLSS  R6, I, 24$                  ; 0820
                                               03 11 001AE          BRB     25$
                                        FF5A   31 001B0 24$:        BRW     10$
                                     009A      C7 9F 001B3 25$:     PUSHAB  P.ABD                                   ; 0846
                                            68 01 FB 001B7          CALLS   #1, OUTPUT_CMDCOUNTED
                                     0D    01  A9 01 E1 001BA 26$:  BBC     #1, DIF$GL_FLAGS+1, 27$                 ; 0850
                                        00000000G   00 DD 001BF     PUSHL   DIF$GL_WIDTH                            ; 0852
                                           009C   C7 9F 001C5      PUSHAB  P.ABE
                                            6A 02 FB 001C9          CALLS   #2, OUTPUT_CMDFAO
                                     0D    01  A9 02 E1 001CC 27$:  BBC     #2, DIF$GL_FLAGS+1, 28$                 ; 0854
                                        00000000G   00 DD 001D1     PUSHL   DIF$GL_MATCH                            ; 0856
                                           00A7   C7 9F 001D7      PUSHAB  P.ABF
                                            6A 02 FB 001DB          CALLS   #2, OUTPUT_CMDFAO
                                     0D    01  A9 06 E1 001DE 28$:  BBC     #6, DIF$GL_FLAGS+1, 29$                 ; 0858
                                        00000000G   00 DD 001E3     PUSHL   DIF$GL_MAXDIF                           ; 0860
                                           00B2   C7 9F 001E9      PUSHAB  P.ABG
                                            6A 02 FB 001ED          CALLS   #2, OUTPUT_CMDFAO
                                     0D        69 05 E1 001F0 29$:  BBC     #5, DIF$GL_FLAGS, 30$                   ; 0862
                                        00000000G   00 DD 001F4     PUSHL   DIF$GL_MERGED                           ; 0864
                                           00CB   C7 9F 001FA      PUSHAB  P.ABH
                                            6A 02 FB 001FE          CALLS   #2, OUTPUT_CMDFAO
                                            08             69 E9 00201 30$:  BLBC    DIF$GL_FLAGS, 31$             ; 0866
                                04          69             01 E0 00204          BBS     #1, DIF$GL_FLAGS, 31$   ; 0867
                                53          69             02 E1 00208          BBC     #2, DIF$GL_FLAGS, 36$   ; 0868
                                               55 D4 0020C 31$:  CLRL    MASK                                    ; 0876
                                            50     0100    C7 D0 0020E          MOVL    LIST-4, I                       ; 0877
                                               10 11 00213          BRB     33$
                                            51     0104 C740 D0 00215 32$:  MOVL    LIST[I], R1                     ; 0879
                                            52             69 D2 0021B          MCOML   DIF$GL_FLAGS, R2
                                51          61             52 CB 0021E          BICL3   R2, (R1), R1
                                            55             51 C8 00222          BISL2   R1, MASK
                                            ED             50 F4 00225 33$:     SOBGEQ  I, 32$
                                                  0110     C7 9F 00228          PUSHAB  P.ABM                           ; 0880
                                            68             01 FB 0022C          CALLS   #1, OUTPUT_CMDCOUNTED
                                            53     0100    C7 D0 0022F          MOVL    LIST-4, I                       ; 0881
                                               1F 11 00234          BRB     35$
                                            50     0104 C743 D0 00236 34$:  MOVL    LIST[I], R0                     ; 0883
                                            55             60 D3 0023C          BITL    (R0), MASK
                                               14 13 0023F          BEQL    35$
                                            55             60 CA 00241          BICL2   (R0), MASK                      ; 0886
                                                  04       A0 9F 00244          PUSHAB  4(R0)                           ; 0887
                                            68             01 FB 00247          CALLS   #1, OUTPUT_CMDCOUNTED
                                                  55 D5 0024A          TSTL    MASK                                    ; 0888
                                               07 13 0024C          BEQL    35$
                                                  0118     C7 9F 0024E          PUSHAB  P.ABN                           ; 0890
                                            68             01 FB 00252          CALLS   #1, OUTPUT_CMDCOUNTED
                                            DE             53 F4 00255 35$:     SOBGEQ  I, 34$                  ; 0883
```

```
                                 011A  C7  9F 00258          PUSHAB  P.ABO                                   ; 0892
                              68       01  FB 0025C          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                                 01    A9  95 0025F 36$:     TSTB    DIF$GL_FLAGS+1                          ; 0895
                                       10  18 00262          BGEQ    37$                                     ;
                                 011C  C7  9F 00264          PUSHAB  P.ABP                                   ; 0898
                              68       01  FB 00268          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                           00000000G   00  9F 0026B          PUSHAB  DIF$GL_OUTDESC                          ; 0899
                              6B       01  FB 00271          CALLS   #1, OUTPUT_CMDENTITY                    ;
                     07       69       06  E1 00274 37$:     BBC     #6, DIF$GL_FLAGS, 38$                   ; 0902
                                 0125  C7  9F 00278          PUSHAB  P.ABQ                                   ; 0904
                              68       01  FB 0027C          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
           50 00000000G       00       02  EF 0027F 38$:     EXTZV   #2, #1, DIF$GL_MASFDB+36, R0            ; 0906
                              17       50  E9 00288          BLBC    R0, 40$                                 ;
                     06 00000000G       00  E1 0028B          BBC     #2, DIF$GL_REVFDB+36, 39$               ;
                                 012F  C7  9F 00293          PUSHAB  P.ABR                                   ; 0908
                                       15  11 00297          BRB     41$                                     ;
                              06       50  E9 00299 39$:     BLBC    R0, 40$                                 ; 0909
                                 013A  C7  9F 0029C          PUSHAB  P.ABS                                   ; 0911
                                       0C  11 002A0          BRB     41$                                     ;
                     07 00000000G       00  E1 002A2 40$:     BBC     #2, DIF$GL_REVFDB+36, 42$               ; 0912
                                 014C  C7  9F 002AA          PUSHAB  P.ABT                                   ; 0914
                              68       01  FB 002AE 41$:     CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                              07       01  A9  E9 002B1 42$:     BLBC    DIF$GL_FLAGS+1, 43$                     ; 0916
                                 0160  C7  9F 002B5          PUSHAB  P.ABU                                   ; 0918
                              68       01  FB 002B9          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                     0D       01       03  E1 002BC 43$:     BBC     #3, DIF$GL_FLAGS+1, 44$                 ; 0920
                           00000000G   00  DD 002C1          PUSHL   DIF$GL_WNDSIZ                           ; 0922
                                 0165  C7  9F 002C7          PUSHAB  P.ABV                                   ;
                              6A       02  FB 002CB          CALLS   #2, OUTPUT_CMDFAO                       ;
                07       01       A9       04  E0 002CE 44$:     BBS     #4, DIF$GL_FLAGS+1, 45$                 ; 0924
                                 0171  C7  9F 002D3          PUSHAB  P.ABW                                   ; 0926
                              68       01  FB 002D7          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                                 017B  C7  9F 002DA 45$:     PUSHAB  P.ABX                                   ; 0928
                              68       01  FB 002DE          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                           00000000G   00  9F 002E1          PUSHAB  DIF$GL_MASFDB                           ; 0930
                     00000000V         EF  01  FB 002E7          CALLS   #1, OUTPUT_CMDFILE                      ;
                                 017C  C7  9F 002EE          PUSHAB  P.ABY                                   ; 0931
                              68       01  FB 002F2          CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                           00000000G   00  9F 002F5          PUSHAB  DIF$GL_REVFDB                           ; 0933
                     00000000V         EF  01  FB 002FB          CALLS   #1, OUTPUT_CMDFILE                      ;
                                       7E  D4 00302          CLRL    -(SP)                                   ; 0934
                              6B       01  FB 00304          CALLS   #1, OUTPUT_CMDENTITY                    ;
                              50       01  D0 00307 46$:     MOVL    #1, R0                                  ; 0938
                                       04  0030A          RET                                             ; 0939

; Routine Size:  779 bytes,    Routine Base:  $CODE$ + 0220
```

```
;  642          0940  1 ROUTINE output_cmdfile (fdb) =
;  643          0941  2 BEGIN
;  644          0942  2
;  645          0943  2 !++
;  646          0944  2 !
;  647          0945  2 ! FUNCTIONAL DESCRIPTION:
;  648          0946  2 !
;  649          0947  2 !       Output a file line in the trailer of the listing.
;  650          0948  2 !
;  651          0949  2 ! INPUTS:
;  652          0950  2 !
;  653          0951  2 !       fdb =   address of file descriptor block for file
;  654          0952  2 !
;  655          0953  2 ! OUTPUTS:
;  656          0954  2 !
;  657          0955  2 !       A file line of the trailer is output.
;  658          0956  2 !
;  659          0957  2 ! ROUTINE VALUES:
;  660          0958  2 !
;  661          0959  2 !       Always true
;  662          0960  2 !
;  663          0961  2 !--
;  664          0962  2 MAP
;  665          0963  2     fdb : REF BBLOCK;                           ! file descriptor block
;  666          0964  2
;  667          0965  2 LOCAL
;  668          0966  2     outdesc : BBLOCK [dsc$c_s_bln];
;  669          0967  2
;  670          0968  2 output_cmdentity (.fdb [fdb$l_fildesc]);
;  671          0969  2 IF .fdb [fdb$v_changebar]
;  672          0970  2 THEN
;  673          0971  3     BEGIN
;  674          0972  3     output_cmdcounted (cstring ('/CHANGE_BAR'));
;  675          0973  4     IF (.fdb [fdb$v_linenum] XOR .dif$gl_flags [dif$v_linenum])
;  676          0974  3         OR .fdb [fdb$b_cbarchr] NEQ %C'!'
;  677          0975  3     THEN
;  678          0976  3         output_cmdcounted (cstring ('='));
;  679          0977  3     IF .fdb [fdb$b_cbarchr] NEQ %C'!'
;  680          0978  3     THEN
;  681          0979  4         BEGIN
;  682          0980  4         outdesc [dsc$w_length] = 1;
;  683          0981  4         outdesc [dsc$a_pointer] = fdb [fdb$b_cbarchr];
;  684          0982  4         output_cmdfao (cstring ('''!AS'''), outdesc);
;  685          0983  3         END;
;  686          0984  4     IF (.fdb [fdb$v_linenum] XOR .dif$gl_flags [dif$v_linenum])
;  687          0985  3         AND .fdb [fdb$b_cbarchr] NEQ %C'!'
;  688          0986  3     THEN
;  689          0987  3         output_cmdcounted (cstring (','));
;  690          0988  3     IF .fdb [fdb$v_linenum] AND NOT .dif$gl_flags [dif$v_linenum]
;  691          0989  3     THEN
;  692          0990  3         output_cmdcounted (cstring ('NUMBER'))
;  693          0991  3     ELSE IF NOT .fdb [fdb$v_linenum] AND .dif$gl_flags [dif$v_linenum]
;  694          0992  3     THEN
;  695          0993  3         output_cmdcounted (cstring ('NONUMBER'));
;  696          0994  4     IF (.fdb [fdb$v_linenum] XOR .dif$gl_flags [dif$v_linenum])
;  697          0995  3         OR .fdb [fdb$b_cbarchr] NEQ %C'!'
;  698          0996  3     THEN
```

```
;  699          0997  3          output_cmdcounted (cstring (')'));
;  700          0998  2      END;
;  701          0999  2
;  702          1000  2 RETURN true;
;  703          1001  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        0B  00209 P.ABZ:  .BYTE   11
                52  41  42  5F  45  47  4E  41  48  43  2F  0020A         .ASCII  \/CHANGE_BAR\                                   ;
                                                        02  00215 P.ACA:  .BYTE   2
                                                    28  3D  00216         .ASCII  \=(\
                                                        05  00218 P.ACB:  .BYTE   5
                                                22  53  41  21  22  00219         .ASCII  \"!AS"\
                                                        01  0021E P.ACC:  .BYTE   1
                                                        2C  0021F         .ASCII  \,\
                                                        06  00220 P.ACD:  .BYTE   6
                                            52  45  42  4D  55  4E  00221         .ASCII  \NUMBER\
                                                        08  00227 P.ACE:  .BYTE   8
                                        52  45  42  4D  55  4E  4F  4E  00228         .ASCII  \NONUMBER\
                                                        01  00230 P.ACF:  .BYTE   1
                                                        29  00231         .ASCII  \)\                                             ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                    00FC 00000 OUTPUT_CMDFILE:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7                          ; 0940
                                        57 00000000G    00  9E  00002         MOVAB   DIF$GL_FLAGS, R7                        ;
                                        56 00000000V    EF  9E  00009         MOVAB   OUTPUT_CMDCOUNTED, R6                   ;
                                        55 00000000'    EF  9E  00010         MOVAB   P.ABZ, R5                               ;
                                        5E              08  C2  00017         SUBL2   #8, SP                                  ;
                                        52          04  AC  D0  0001A         MOVL    FDB, R2                                 ; 0968
                                                    38  A2  DD  0001E         PUSHL   56(R2)                                  ;
                                00000000V   EF          01  FB  00021         CALLS   #1, OUTPUT_CMDENTITY                    ;
                                        03          24  A2  E8  00028         BLBS    36(R2), 1$                              ; 0969
                                                0086    31  0002C         BRW     11$                                     ;
                                                        55  DD  0002F 1$:     PUSHL   R5                                      ; 0972
                                        66              01  FB  00031         CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                53      24  A2          01              01  EF  00034         EXTZV   #1, #1, 36(R2), R3                      ; 0973
                50      01  A7          01              04  EF  0003A         EXTZV   #4, #1, DIF$GL_FLAGS+1, R0              ;
                                        50              53  C0  00040         ADDL2   R3, R0                                  ;
                                        06              50  E8  00043         BLBS    R0, 2$                                  ;
                                        21          26  A2  91  00046         CMPB    38(R2), #33                             ; 0974
                                                        06  13  0004A         BEQL    3$                                      ;
                                                    0C  A5  9F  0004C 2$:     PUSHAB  P.ACA                                   ; 0976
                                        66              01  FB  0004F         CALLS   #1, OUTPUT_CMDCOUNTED                   ;
                                                        54  D4  00052 3$:     CLRL    R4                                      ; 0977
                                        21          26  A2  91  00054         CMPB    38(R2), #33                             ;
                                                        16  13  00058         BEQL    4$                                      ;
                                                        54  D6  0005A         INCL    R4                                      ;
                                        6E              01  B0  0005C         MOVW    #1, OUTDESC                             ; 0980
                                    04  AE          26  A2  9E  0005F         MOVAB   38(R2), OUTDESC+4                       ; 0981
                                                        5E  DD  00064         PUSHL   SP                                      ; 0982
```

```
                                        0F   A5   9F 00066       PUSHAB  P.ACB                            ;
                       00000000V   EF        02   FB 00069       CALLS   #2, OUTPUT_CMDFAO                ;
     50        01   A7             01        04   EF 00070 4$:   EXTZV   #4, #1, DIF$GL_FLAGS+1, R0       ; 0984
                                   50        53   C0 00076       ADDL2   R3, R0                           ;
                                   09        50   E9 00079       BLBC    R0, 5$                           ;
                                   06        54   E9 0007C       BLBC    R4, 5$                           ; 0985
                                        15   A5   9F 0007F       PUSHAB  P.ACC                            ; 0987
                                   66        01   FB 00082       CALLS   #1, OUTPUT_CMDCOUNTED            ;
                                   0D        53   E9 00085 5$:   BLBC    R3, 7$                           ; 0988
                    05        01   A7        04   E0 00088       BBS     #4, DIF$GL_FLAGS+1, 6$           ;
                                        17   A5   9F 0008D       PUSHAB  P.ACD                            ; 0990
                                             0B   11 00090       BRB     8$                               ;
                                   0B        53   E8 00092 6$:   BLBS    R3, 9$                           ; 0991
                    06        01   A7        04   E1 00095 7$:   BBC     #4, DIF$GL_FLAGS+1, 9$           ;
                                        1E   A5   9F 0009A       PUSHAB  P.ACE                            ; 0993
                                   66        01   FB 0009D 8$:   CALLS   #1, OUTPUT_CMDCOUNTED            ;
     50        01   A7             01        04   EF 000A0 9$:   EXTZV   #4, #1, DIF$GL_FLAGS+1, R0       ; 0994
                                   50        53   C0 000A6       ADDL2   R3, R0                           ;
                                   03        50   E8 000A9       BLBS    R0, 10$                          ;
                                   06        54   E9 000AC       BLBC    R4, 11$                          ; 0995
                                        27   A5   9F 000AF 10$:  PUSHAB  P.ACF                            ; 0997
                                   66        01   FB 000B2       CALLS   #1, OUTPUT_CMDCOUNTED            ;
                                   50        01   D0 000B5 11$:  MOVL    #1, R0                           ; 1000
                                                  04 000B8       RET                                      ; 1001

; Routine Size:  185 bytes,    Routine Base:  $CODE$ + 052B
```

```
;   705          1002  1 ROUTINE output_cmdfao (control, data1) =
;   706          1003  2 BEGIN
;   707          1004  2
;   708          1005  2 !++
;   709          1006  2 !
;   710          1007  2 ! FUNCTIONAL DESCRIPTION:
;   711          1008  2 !
;   712          1009  2 !       Output part of trail using $FAO with one argument.
;   713          1010  2 !
;   714          1011  2 ! INPUTS:
;   715          1012  2 !
;   716          1013  2 !       control =       $FAO control string (counted string)
;   717          1014  2 !       data1   =       $FAO data item number 1
;   718          1015  2 !
;   719          1016  2 ! OUTPUTS:
;   720          1017  2 !
;   721          1018  2 !       Part of the trailer is output.
;   722          1019  2 !
;   723          1020  2 ! ROUTINE VALUES:
;   724          1021  2 !
;   725          1022  2 !       Always true
;   726          1023  2 !
;   727          1024  2 !--
;   728          1025  2 MAP
;   729          1026  2     control : REF VECTOR [, BYTE];
;   730          1027  2
;   731          1028  2 LOCAL
;   732          1029  2     ctrdesc : BBLOCK [dsc$c_s_bln],             ! control string descriptor
;   733          1030  2     outdesc : BBLOCK [dsc$c_s_bln],
;   734          1031  2     tmpbuf : VECTOR [nam$c_maxrss+dif$c_maxlisiz, BYTE];        ! formatting buffer
;   735          1032  2
;   736          1033  2 ctrdesc [dsc$w_length] = .control [0];
;   737          1034  2 ctrdesc [dsc$a_pointer] = control [1];
;   738          1035  2 outdesc [dsc$w_length] = %ALLOCATION (tmpbuf);
;   739          1036  2 outdesc [dsc$a_pointer] = tmpbuf;
;   740          1037  2 SYS$FAO (ctrdesc, outdesc [dsc$w_length], outdesc, .data1);
;   741          1038  2 output_cmdentity (outdesc);
;   742          1039  2
;   743          1040  2 RETURN true;
;   744          1041  1 END;


                                0000 00000 OUTPUT_CMDFAO:
                                                .WORD   Save nothing                    ; 1002
                        5E      FE6C    CE  9E 00002    MOVAB   -404(SP), SP            ;
                    F8  AD        04    BC  9B 00007    MOVZBW  @CONTROL, CTRDESC       ; 1033
FC  AD              04  AC              01  C1 0000C    ADDL3   #1, CONTROL, CTRDESC+4  ; 1034
                    F0  AD      0183    8F  B0 00012    MOVW    #387, OUTDESC           ; 1035
                    F4  AD              6E  9E 00018    MOVAB   TMPBUF, OUTDESC+4       ; 1036
                          08    AC    DD  9F 0001C    PUSHL   DATA1                   ; 1037
                          F0    AD    9F  9F 0001F    PUSHAB  OUTDESC                 ;
                          F0    AD    9F  9F 00022    PUSHAB  OUTDESC                 ;
                          F8    AD    9F  9F 00025    PUSHAB  CTRDESC                 ;
        00000000G   00                  04  FB 00028    CALLS   #4, SYS$FAO             ;
```

```
                         FO   AD  9F 0002F         PUSHAB  OUTDESC                                  ; 1038
                00000000V EF      01  FB 00032     CALLS   #1, OUTPUT_CMDENTITY                    ;
                         50       01  D0 00039     MOVL    #1, R0                                  ; 1040
                                      04 0003C     RET                                             ; 1041

; Routine Size:  61 bytes,    Routine Base:  $CODE$ + 05E4
```

```
;  746          1042  1 ROUTINE output_cmdcounted (str) =
;  747          1043  2 BEGIN
;  748          1044  2
;  749          1045  2 !++
;  750          1046  2 ! FUNCTIONAL DESCRIPTION:
;  751          1047  2 !
;  752          1048  2 !       Output a command line entity.  Insert new
;  753          1049  2 !       lines as necessary.
;  754          1050  2 !
;  755          1051  2 ! INPUTS:
;  756          1052  2 !
;  757          1053  2 !       str =           Counted string to output
;  758          1054  2 !
;  759          1055  2 ! OUTPUTS:
;  760          1056  2 !
;  761          1057  2 !       The entity is inserted in the output buffer.  Output buffers
;  762          1058  2 !       are written as required.
;  763          1059  2 !
;  764          1060  2 ! IMPLICIT INPUTS/OUTPUTS:
;  765          1061  2 !
;  766          1062  2 !       cmd_bufpos =    current position in output buffer
;  767          1063  2 !
;  768          1064  2 ! ROUTINE VALUES:
;  769          1065  2 !
;  770          1066  2 !       Always true.
;  771          1067  2 !
;  772          1068  2 !--
;  773          1069  2 MAP
;  774          1070  2     str : REF VECTOR [, BYTE];                  ! counted string
;  775          1071  2
;  776          1072  2 LOCAL
;  777          1073  2     outdesc : BBLOCK [dsc$c_s_bln];             ! string descriptor
;  778          1074  2
;  779          1075  2 outdesc [dsc$w_length] = .str [0];
;  780          1076  2 outdesc [dsc$a_pointer] = str [1];
;  781          1077  2 output_cmdentity (outdesc);
;  782          1078  2
;  783          1079  2 RETURN true;
;  784          1080  2 END;


                                        0000 00000 OUTPUT_CMDCOUNTED:
                                                   .WORD    Save nothing                    ; 1042
                                5E      08   C2 00002      SUBL2    #8, SP                  ;
                                6E      04   BC 9B 00005   MOVZBW   @STR, OUTDESC          ; 1076
                    04   AE        04    AC   01 C1 00009 ADDL3    #1, STR, OUTDESC+4      ; 1077
                                          5E   DD 0000F    PUSHL    SP                      ; 1078
                         00000000V  EF         01 FB 00011 CALLS    #1, OUTPUT_CMDENTITY    ;
                                50         01   D0 00018   MOVL     #1, R0                  ; 1080
                                           04 0001B        RET                              ; 1081

; Routine Size:  28 bytes,    Routine Base:  $CODE$ + 0621
```

```
;  787          1082  1 ROUTINE output_cmdentity (descr) =
;  788          1083  2 BEGIN
;  789          1084  2
;  790          1085  2 !++
;  791          1086  2 !
;  792          1087  2 ! FUNCTIONAL DESCRIPTION:
;  793          1088  2 !
;  794          1089  2 !       Output a command line entity.  Insert new
;  795          1090  2 !       lines as necessary.
;  796          1091  2 !
;  797          1092  2 ! INPUTS:
;  798          1093  2 !
;  799          1094  2 !       descr =         descriptor of entity to output
;  800          1095  2 !                       If descr = 0, output buffer
;  801          1096  2 !                       If length = 0, output buffer with continuation mark
;  802          1097  2 !                       Otherwise, buffer and output data
;  803          1098  2 !
;  804          1099  2 ! OUTPUTS:
;  805          1100  2 !
;  806          1101  2 !       The entity is inserted in the output buffer.  Output buffers
;  807          1102  2 !       are written as required.
;  808          1103  2 !
;  809          1104  2 ! IMPLICIT INPUTS/OUTPUTS:
;  810          1105  2 !
;  811          1106  2 !       cmd_bufpos =    current position in output buffer
;  812          1107  2 !
;  813          1108  2 ! ROUTINE VALUES:
;  814          1109  2 !
;  815          1110  2 !       Always true.
;  816          1111  2 !
;  817          1112  2 !--
;  818          1113  2 MAP
;  819          1114  2     descr : REF BBLOCK;                         ! string descriptor
;  820          1115  2
;  821          1116  2 LOCAL
;  822          1117  2     outdesc : BBLOCK [dsc$c_s_bln];             ! string descriptor
;  823          1118  2
;  824          1119  2 LITERAL
;  825          1120  2     indent = MINU (4, dif$c_minlisiz-1);        ! indentation of continuation line
;  826          1121  2
;  827          1122  2 IF .descr EQL 0
;  828          1123  2 THEN
;  829          1124  3     BEGIN                                       ! flush output buffer
;  830          1125  3     outdesc [dsc$w_length] = .cmd_bufpos;
;  831          1126  3     outdesc [dsc$a_pointer] = .dif$gl_outbuf;
;  832          1127  3     put_desc (outdesc);
;  833          1128  3     cmd_bufpos = 0;
;  834          1129  3     END
;  835          1130  2 ELSE IF .descr [dsc$w_length] EQL 0
;  836          1131  2 THEN
;  837          1132  3     BEGIN                                   ! output line with continuation
;  838          1133  3     dif$gl_outbuf [.cmd_bufpos] = %C'-';
;  839          1134  3     cmd_bufpos = .cmd_bufpos+1;
;  840          1135  3     output_cmdentity (0);                   ! output line
;  841          1136  3     CH$FILL (%C' ', indent, .dif$gl_outbuf);
;  842          1137  3     cmd_bufpos = indent;
;  843          1138  3     END
```

```
;  844          1139  2 ELSE IF .descr [dsc$w_length] LEQU .dif$gl_width-.cmd_bufpos-1
;  845          1140  2 THEN
;  846          1141  3     BEGIN                                   ! fits on current line
;  847          1142  3     CH$MOVE (.descr [dsc$w_length], .descr [dsc$a_pointer], dif$gl_outbuf [.cmd_bufpos]);
;  848          1143  3     cmd_bufpos = .cmd_bufpos+.descr [dsc$w_length];
;  849          1144  3     END
;  850          1145  2 ELSE IF .descr [dsc$w_length] LEQU .dif$gl_width-indent-1
;  851          1146  2 THEN
;  852          1147  3     BEGIN                                   ! output line and put on new line
;  853          1148  3     outdesc [dsc$w_length] = 0;
;  854          1149  3     output_cmdentity (outdesc);             ! output line with continuation
;  855          1150  3     output_cmdentity (.descr);              ! put entity on new line
;  856          1151  3     END
;  857          1152  2 ELSE                                       ! item too big for one line
;  858          1153  3     BEGIN
;  859          1154  3     IF .dif$gl_width-.cmd_bufpos-1 EQL 0
;  860          1155  3     THEN
;  861          1156  4         BEGIN
;  862          1157  4         outdesc [dsc$w_length] = 0;
;  863          1158  4         output_cmdentity (outdesc);
;  864          1159  3         END;
;  865          1160  3     outdesc [dsc$a_pointer] = .descr [dsc$a_pointer];
;  866          1161  3     WHILE true
;  867          1162  3     DO
;  868          1163  4         BEGIN
;  869          1164  4         outdesc [dsc$w_length] = MINU (
;  870          1165  4             .dif$gl_width-.cmd_bufpos-1,
;  871          1166  4             .descr [dsc$a_pointer]+.descr [dsc$w_length]-.outdesc [dsc$a_pointer]);
;  872          1167  4         output_cmdentity (outdesc);
;  873          1168  4         outdesc [dsc$a_pointer] = .outdesc [dsc$a_pointer]+.outdesc [dsc$w_length];
;  874          1169  4         IF .outdesc [dsc$a_pointer] EQLA .descr [dsc$a_pointer]+.descr [dsc$w_length]
;  875          1170  4         THEN
;  876          1171  4             EXITLOOP;
;  877          1172  4         outdesc [dsc$w_length] = 0;
;  878          1173  4         output_cmdentity (outdesc);         ! output line with continuation
;  879          1174  4         cmd_bufpos = 0;                     ! no indentation
;  880          1175  3         END;
;  881          1176  2     END;
;  882          1177  2
;  883          1178  2 RETURN true;
;  884          1179  1 END;


                                OFFC 00000 OUTPUT_CMDENTITY:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1082
5B 00000000G   00   9E 00002               MOVAB   DIF$GL_WIDTH, R11                             ;
5A 00000000G   00   9E 00009               MOVAB   DIF$GL_OUTBUF, R10                            ;
59         ED  AF   9E 00010               MOVAB   OUTPUT_CMDENTITY, R9                          ;
58 00000000'   EF   9E 00014               MOVAB   CMD_BUFPOS, R8                                ;
5E             08   C2 0001B               SUBL2   #8, SP                                        ;
52             68   D0 0001E               MOVL    CMD_BUFPOS, R2                                ; 1125
56         04  AC   D0 00021               MOVL    DESCR, R6                                     ; 1122
                14   12 00025               BNEQ    1$                                            ;
6E             52   B0 00027               MOVW    R2, OUTDESC                                   ; 1125
```

```
                        04   AE          6A  D0 0002A          MOVL    DIF$GL_OUTBUF, OUTDESC+4                        ; 1126
                                         5E  DD 0002E          PUSHL   SP                                              ; 1127
                00000000V    EF          01  FB 00030          CALLS   #1, PUT_DESC
                                         68  D4 00037          CLRL    CMD_BUFPOS                                      ; 1128
                                         5B  11 00039          BRB     4$                                              ; 1122
                                         66  B5 0003B 1$:       TSTW    (R6)                                            ; 1130
                                         1D  12 0003D          BNEQ    2$
                          50             6A  D0 0003F          MOVL    DIF$GL_OUTBUF, R0                               ; 1133
                        6240             2D  90 00042          MOVB    #45, (R2)[R0]
                                         68  D6 00046          INCL    CMD_BUFPOS                                      ; 1134
                                         7E  D4 00048          CLRL    -(SP)                                           ; 1135
                          69             01  FB 0004A          CALLS   #1, OUTPUT_CMDENTITY
                          50             6A  D0 0004D          MOVL    DIF$GL_OUTBUF, R0                               ; 1136
                          60 20202020    8F  D0 00050          MOVL    #538976288, (R0)
                          68             04  D0 00057          MOVL    #4, CMD_BUFPOS                                  ; 1137
                                         3A  11 0005A          BRB     4$                                              ; 1130
                          51             6B  D0 0005C 2$:       MOVL    DIF$GL_WIDTH, R1                                ; 1139
                    57    51             52  C3 0005F          SUBL3   R2, R1, R7
                          50       FF    A7  9E 00063          MOVAB   -1(R7), R0
                50        66             10  00  ED 00067      CMPZV   #0, #16, (R6), R0
                                         11  1A 0006C          BGTRU   3$
                          50             6A  D0 0006E          MOVL    DIF$GL_OUTBUF, R0                               ; 1142
                  6240    04   B6        66  28 00071          MOVC3   (R6), @4(R6), (R2)[R0]
                          50             66  3C 00077          MOVZWL  (R6), R0                                        ; 1143
                          68             50  C0 0007A          ADDL2   R0, CMD_BUFPOS
                                         64  11 0007D          BRB     9$                                              ; 1139
                          50       FB    A1  9E 0007F 3$:       MOVAB   -5(R1), R0                                      ; 1145
                50        66             10  00  ED 00083      CMPZV   #0, #16, (R6), R0
                                         0E  1A 00088          BGTRU   5$
                                         6E  B4 0008A          CLRW    OUTDESC                                         ; 1148
                                         5E  DD 0008C          PUSHL   SP                                              ; 1149
                          69             01  FB 0008E          CALLS   #1, OUTPUT_CMDENTITY
                                         56  DD 00091          PUSHL   R6                                              ; 1150
                          69             01  FB 00093          CALLS   #1, OUTPUT_CMDENTITY
                                         4B  11 00096 4$:       BRB     9$                                              ; 1145
                          01             57  D1 00098 5$:       CMPL    R7, #1                                          ; 1154
                                         07  12 0009B          BNEQ    6$
                                         6E  B4 0009D          CLRW    OUTDESC                                         ; 1157
                                         5E  DD 0009F          PUSHL   SP                                              ; 1158
                          69             01  FB 000A1          CALLS   #1, OUTPUT_CMDENTITY
                    04    AE       04    A6  D0 000A4 6$:       MOVL    4(R6), OUTDESC+4                                ; 1160
                51        6B             68  C3 000A9 7$:       SUBL3   CMD_BUFPOS, DIF$GL_WIDTH, R1                    ; 1165
                                         51  D7 000AD          DECL    R1
                          52             66  3C 000AF          MOVZWL  (R6), R2                                        ; 1166
                          52       04    A6  C0 000B2          ADDL2   4(R6), R2
                50        52       04    AE  C3 000B6          SUBL3   OUTDESC+4, R2, R0
                          50             51  D1 000BB          CMPL    R1, R0
                                         03  1B 000BE          BLEQU   8$
                          51             50  D0 000C0          MOVL    R0, R1
                          6E             51  B0 000C3 8$:       MOVW    R1, OUTDESC                                     ; 1164
                                         5E  DD 000C6          PUSHL   SP                                              ; 1167
                          69             01  FB 000C8          CALLS   #1, OUTPUT_CMDENTITY
                          50             6E  3C 000CB          MOVZWL  OUTDESC, R0                                     ; 1168
                    04    AE             50  C0 000CE          ADDL2   R0, OUTDESC+4
                          52       04    AE  D1 000D2          CMPL    OUTDESC+4, R2                                   ; 1169
                                         0B  13 000D6          BEQL    9$
                                         6E  B4 000D8          CLRW    OUTDESC                                         ; 1172
```

```
                          5E  DD 000DA          PUSHL   SP                                                      ; 1173
                      69  01  FB 000DC          CALLS   #1, OUTPUT_CMDENTITY                                    ;
                          68  D4 000DF          CLRL    CMD_BUFPOS                                              ; 1174
                          C6  11 000E1          BRB     7$                                                      ; 1161
                      50  01  D0 000E3 9$:      MOVL    #1, R0                                                  ; 1178
                          04 000E6              RET                                                             ; 1179

; Routine Size:  231 bytes,    Routine Base:  $CODE$ + 063D
```

```
;  886     1180  1 ROUTINE output_changebar (fdb)  =
;  887     1181  2 BEGIN
;  888     1182  2
;  889     1183  2 !++
;  890     1184  2 !
;  891     1185  2 ! FUNCTIONAL DESCRIPTION:
;  892     1186  2 !
;  893     1187  2 !       This routine is called to output the most recent set of records,
;  894     1188  2 !       read from a particular input file, in CHANGEBAR format.
;  895     1189  2 !
;  896     1190  2 ! INPUTS:
;  897     1191  2 !
;  898     1192  2 !       fdb =   The address of the FDB of the desired input file.
;  899     1193  2 !
;  900     1194  2 ! OUTPUTS:
;  901     1195  2 !
;  902     1196  2 !       The differences are written in changebar format to the output file.
;  903     1197  2 !
;  904     1198  2 ! ROUTINE VALUES:
;  905     1199  2 !
;  906     1200  2 !       Always true
;  907     1201  2 !
;  908     1202  2 !--
;  909     1203  2
;  910     1204  2 MAP
;  911     1205  2     fdb : REF BBLOCK;
;  912     1206  2
;  913     1207  2 LOCAL
;  914     1208  2     cbarflag,                       ! Flag to output a change bar
;  915     1209  2     prevmatch,                      ! Flag is true if last record was a match
;  916     1210  2     rdb : REF BBLOCK;               ! Address of the RDB of the current record
;  917     1211  2
;  918     1212  2 rdb = .fdb [fdb$l_firstdif];                    ! Get first difference record
;  919     1213  2 prevmatch = true;                               ! Assume last record was a match
;  920     1214  2
;  921     1215  3 WHILE (.rdb NEQ .fdb [fdb$l_compnrec])          ! Output all unmatched records
;  922     1216  3 DO BEGIN
;  923     1217  3    IF .prevmatch AND .rdb [rdb$v_matchone]      ! Ouput change bar if difference or if
;  924     1218  3       THEN cbarflag = 1                        ! first match after a deletion
;  925     1219  3       ELSE cbarflag = (IF .rdb [rdb$v_match] THEN 2 ELSE 1);
;  926     1220  3    fdb [fdb$l_currec] = .rdb;                   ! Specify RDB of output record
;  927     1221  3    put_record (.fdb, .cbarflag);                ! Output the record
;  928     1222  3    IF NOT .rdb [rdb$v_ignored]                  ! If record was not ignored
;  929     1223  3       THEN prevmatch = .rdb [rdb$v_match];      ! Update previous match flag
;  930     1224  3    rdb = .rdb [rdb$l_flink];                    ! Get next record
;  931     1225  2    END;
;  932     1226  2
;  933     1227  2 IF .prevmatch AND .rdb [rdb$v_matchone]          ! Output change bar if first match
;  934     1228  2    THEN cbarflag = 1                            ! after a deletion
;  935     1229  2    ELSE cbarflag = (IF .rdb [rdb$v_match] THEN 2 ELSE 1);
;  936     1230  2 fdb [fdb$l_currec] = .rdb;                      ! Specify RDB of output record
;  937     1231  2 put_record (.fdb, .cbarflag);                   ! Output the record
;  938     1232  2
;  939     1233  2 RETURN true;
;  940     1234  1 END;
```

```
                                007C 00000 OUTPUT_CHANGEBAR:
                                           .WORD   Save R2,R3,R4,R5,R6                                 ; 1180
                          56 00000000V  EF  9E 00002       MOVAB   PUT_RECORD, R6                      ;
                          53       04   AC  D0 00009       MOVL    FDB, R3                             ; 1212
                          52       0C   A3  D0 0000D       MOVL    12(R3), RDB                         ;
                          55            01  D0 00011       MOVL    #1, PREVMATCH                       ; 1213
                      14  A3            52  D1 00014 1$:    CMPL    RDB, 20(R3)                         ; 1215
                                        2C  13 00018       BEQL    6$                                  ;
                          05            55  E9 0001A       BLBC    PREVMATCH, 2$                       ; 1217
                    0A    08  A2        05  E0 0001D       BBS     #5, 8(RDB), 3$                      ;
                    05    08  A2        04  E1 00022 2$:    BBC     #4, 8(RDB), 3$                      ; 1219
                          54            02  D0 00027       MOVL    #2, CBARFLAG                        ;
                                        03  11 0002A       BRB     4$                                  ;
                          54            01  D0 0002C 3$:    MOVL    #1, CBARFLAG                        ;
                          63            52  D0 0002F 4$:    MOVL    RDB, (R3)                           ; 1220
                                        18  BB 00032       PUSHR   #^M<R3,R4>                          ; 1221
                          66            02  FB 00034       CALLS   #2, PUT_RECORD                      ;
                          06       08   A2  E8 00037       BLBS    8(RDB), 5$                          ; 1222
55                  08    A2            01  04 EF 0003B       EXTZV   #4, #1, 8(RDB), PREVMATCH           ; 1223
                          52            62  D0 00041 5$:    MOVL    (RDB), RDB                          ; 1224
                                        CE  11 00044       BRB     1$                                  ; 1215
                          05            55  E9 00046 6$:    BLBC    PREVMATCH, 7$                       ; 1227
                    0A    08  A2        05  E0 00049       BBS     #5, 8(RDB), 8$                      ;
                    05    08  A2        04  E1 0004E 7$:    BBC     #4, 8(RDB), 8$                      ; 1229
                          54            02  D0 00053       MOVL    #2, CBARFLAG                        ;
                                        03  11 00056       BRB     9$                                  ;
                          54            01  D0 00058 8$:    MOVL    #1, CBARFLAG                        ;
                          63            52  D0 0005B 9$:    MOVL    RDB, (R3)                           ; 1230
                                        18  BB 0005E       PUSHR   #^M<R3,R4>                          ; 1231
                          66            02  FB 00060       CALLS   #2, PUT_RECORD                      ;
                          50            01  D0 00063       MOVL    #1, R0                              ; 1233
                                        04  00066       RET                                         ; 1234

; Routine Size:  103 bytes,    Routine Base:  $CODE$ + 0724
```

```
;  942     1235  1 ROUTINE output_merged =
;  943     1236  2 BEGIN
;  944     1237  2
;  945     1238  2 !++
;  946     1239  2 !
;  947     1240  2 ! FUNCTIONAL DESCRIPTION:
;  948     1241  2 !
;  949     1242  2 !       This routine is called to output the most recent set of detected
;  950     1243  2 !       differences in MERGED format.
;  951     1244  2 !
;  952     1245  2 ! IMPLICIT INPUTS:
;  953     1246  2 !
;  954     1247  2 !       The FDB's of the master and revision files.
;  955     1248  2 !
;  956     1249  2 ! OUTPUTS:
;  957     1250  2 !
;  958     1251  2 !       The differences are written to the output file in merged format.
;  959     1252  2 !
;  960     1253  2 ! ROUTINE VALUES:
;  961     1254  2 !
;  962     1255  2 !       Always true
;  963     1256  2 !
;  964     1257  2 !--
;  965     1258  2
;  966     1259  2 LOCAL
;  967     1260  2     done,                                   ! Flag is set when all differences have been output
;  968     1261  2     fdb : REF BBLOCK,                       ! Address of FDB of current output source
;  969     1262  2     rdb : REF BBLOCK,                       ! Address of RDB of current record
;  970     1263  2     stardesc : BBLOCK [dsc$c_s_bln];        ! Descriptor for line of stars
;  971     1264  2
;  972     1265  2 IF .dif$gl_flags [dif$v_init]               ! If init flag is set
;  973     1266  2    THEN dif$gl_flags [dif$v_init] = false;  ! Then reset flag
;  974     1267  2
;  975     1268  2 done = false;                               ! Init until flag
;  976     1269  2 stardesc [dsc$a_pointer] = stars [1];       ! Init star desc address field
;  977     1270  2 stardesc [dsc$w_length] = .stars [0];       ! Init star desc length field
;  978     1271  2 dif$gl_masfdb [fdb$l_comprec] = .dif$gl_masfdb [fdb$l_firstdif]; ! Get first unmatched record
;  979     1272  2 dif$gl_revfdb [fdb$l_comprec] = .dif$gl_revfdb [fdb$l_firstdif]; ! from each input file
;  980     1273  2
;  981     1274  3 DO BEGIN                                   ! Loop until all differences output
;  982     1275  3
;  983     1276  3 fdb = dif$gl_masfdb;                       ! Output master file differences first
;  984     1277  3
;  985     1278  3 INCR i FROM 1 TO 2                         ! For both input files
;  986     1279  4 DO BEGIN                                   ! Output differences
;  987     1280  4
;  988     1281  4    put_desc (stardesc);                    ! Output line of stars
;  989     1282  4    put_idline (.fdb);                      ! Output file id line
;  990     1283  4    rdb = .fdb [fdb$l_comprec];
;  991     1284  4
;  992     1285  5    WHILE ((NOT .rdb [rdb$v_matchone]) AND  ! Output all unmatched records
;  993     1286  5          (NOT .rdb [rdb$v_eof]))
;  994     1287  5    DO BEGIN
;  995     1288  5       IF NOT .rdb [rdb$v_match]            ! If not a match record
;  996     1289  6          THEN BEGIN                        ! Then output it
;  997     1290  6               fdb [fdb$l_currec] = .rdb;   ! Specify record to output
;  998     1291  6               put_record (.fdb, 0);        ! Output record
```

```
;  999     1292  5                END;
; 1000     1293  5            rdb = .rdb [rdb$l_flink];                   ! Get next record
; 1001     1294  4            END;
; 1002     1295  4
; 1003     1296  4        INCR j FROM 1 TO .dif$gl_merged                 ! Output specified number of trailing
; 1004     1297  5        DO BEGIN                                        ! matched records
; 1005     1298  5
; 1006     1299  5            IF .rdb EQL .fdb [fdb$l_compnrec]           ! Check to see if no more differences
; 1007     1300  5                THEN done = true;                       ! No more, then set flag
; 1008     1301  5
; 1009     1302  5            IF .rdb [rdb$v_ignored]                     ! If ignore, then can't be a match
; 1010     1303  5                THEN j = j-1                            ! Look again for a matched record
; 1011     1304  6                ELSE BEGIN                              ! Match, so output record
; 1012     1305  6                     fdb [fdb$l_currec] = .rdb;         ! Specify record to output
; 1013     1306  6                     put_record (.fdb, 0);              ! Output record
; 1014     1307  5                     END;
; 1015     1308  5
; 1016     1309  5            rdb = .rdb [rdb$l_flink];                   ! Get next record
; 1017     1310  4            END;
; 1018     1311  4
; 1019     1312  4        IF .rdb EQL .fdb [fdb$l_compnrec]               ! Check to see if no more differences
; 1020     1313  4            THEN done = true;                           ! No more, then set flag
; 1021     1314  4
; 1022     1315  4        IF .dif$gl_merged EQL 0                         ! If no match records should be output
; 1023     1316  4            THEN rdb = .rdb [rdb$l_flink];              ! Then skip one anyway
; 1024     1317  4
; 1025     1318  4        fdb [fdb$l_comp1rec] = .rdb;                    ! Respecify first record of next potential
; 1026     1319  4                                                        ! difference section
; 1027     1320  4        fdb = dif$gl_revfdb;                            ! Get revision file fdb
; 1028     1321  4        stardesc [dsc$w_length] = .stars [0] / 2;       ! Use fewer stars
; 1029     1322  3        END;                                            ! Of increment
; 1030     1323  3
; 1031     1324  3    stardesc [dsc$w_length] = .stars [0];               ! Init star desc length field
; 1032     1325  3    put_desc (stardesc);                                ! Output stars
; 1033     1326  3
; 1034     1327  3    END                                                 ! Of Until
; 1035     1328  2    UNTIL (.done);
; 1036     1329  2
; 1037     1330  2    RETURN true;
; 1038     1331  1    END;                                                ! Of output_merged


                                    0FFC 00000 OUTPUT_MERGED:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 1235
                         5B 00000000V   EF   9E 00002          MOVAB   PUT_RECORD, R11                     ;
                         5A 00000000V   EF   9E 00009          MOVAB   PUT_DESC, R10                       ;
                         59 00000000G   00   9E 00010          MOVAB   DIF$GL_REVFDB+12, R9                ;
                         58 00000000G   00   9E 00017          MOVAB   DIF$GL_MASFDB+12, R8                ;
                         57 00000000'   EF   9E 0001E          MOVAB   STARS, R7                           ;
                         5E             0C   C2 00025          SUBL2   #12, SP                             ;
                07 00000000G   00       05   E1 00028          BBC     #5, DIF$GL_FLAGS+1, 1$              ; 1265
                   00000000G   00       20   8A 00030          BICB2   #32, DIF$GL_FLAGS+1                 ; 1266
                                        55   D4 00037 1$:       CLRL    DONE                                ; 1268
                         08    AE   01   A7   9E 00039          MOVAB   STARS+1, STARDESC+4                 ; 1269
```

```
                04   AE          67  9B 0003E          MOVZBW  STARS, STARDESC                         ; 1270
                04   A8          68  D0 00042          MOVL    DIF$GL_MASFDB+12, DIF$GL_MASFDB+16      ; 1271
                04   A9          69  D0 00046          MOVL    DIF$GL_REVFDB+12, DIF$GL_REVFDB+16      ; 1272
                     53     F4   A8  9E 0004A 2$:      MOVAB   DIF$GL_MASFDB, FDB                     ; 1276
                     56          01  D0 0004E          MOVL    #1, I                                   ; 1278
                     04     AE   9F  00051 3$:         PUSHAB  STARDESC                                ; 1281
                     6A          01  FB 00054          CALLS   #1, PUT_DESC
                                 53  DD 00057          PUSHL   FDB                                     ; 1282
          00000000V  EF          01  FB 00059          CALLS   #1, PUT_IDLINE
                     52     10   A3  D0 00060          MOVL    16(FDB), RDB                            ; 1283
          19    08   A2          05  E0 00064 4$:      BBS     #5, 8(RDB), 6$                          ; 1285
          14    08   A2          02  E0 00069          BBS     #2, 8(RDB), 6$                          ; 1286
          0A    08   A2          04  E0 0006E          BBS     #4, 8(RDB), 5$                          ; 1288
                     63          52  D0 00073          MOVL    RDB, (FDB)                              ; 1290
                                 7E  D4 00076          CLRL    -(SP)                                   ; 1291
                                 53  DD 00078          PUSHL   FDB
                     6B          02  FB 0007A          CALLS   #2, PUT_RECORD
                     52          62  D0 0007D 5$:      MOVL    (RDB), RDB                              ; 1293
                                 E2  11 00080          BRB     4$                                      ; 1285
                     54 00000000G 00  D0 00082 6$:      MOVL    DIF$GL_MERGED, R4                       ; 1296
                                 6E  D4 00089          CLRL    J                                       ; 1299
                                 20  11 0008B          BRB     11$
                14   A3          52  D1 0008D 7$:      CMPL    RDB, 20(FDB)
                                 03  12 00091          BNEQ    8$
                     55          01  D0 00093          MOVL    #1, DONE                                ; 1300
                     06     08   A2  E9 00096 8$:      BLBC    8(RDB), 9$                              ; 1302
                     6E     FF   AE  9E 0009A          MOVAB   J-1, J                                  ; 1303
                                 0A  11 0009E          BRB     10$
                     63          52  D0 000A0 9$:      MOVL    RDB, (FDB)                              ; 1305
                                 7E  D4 000A3          CLRL    -(SP)                                   ; 1306
                                 53  DD 000A5          PUSHL   FDB
                     6B          02  FB 000A7          CALLS   #2, PUT_RECORD
                     52          62  D0 000AA 10$:     MOVL    (RDB), RDB                              ; 1309
          DC         6E          54  F3 000AD 11$:     AOBLEQ  R4, J, 7$                               ; 1296
                14   A3          52  D1 000B1          CMPL    RDB, 20(FDB)                            ; 1312
                                 03  12 000B5          BNEQ    12$
                     55          01  D0 000B7          MOVL    #1, DONE                                ; 1313
                           00000000G 00  D5 000BA 12$:     TSTL    DIF$GL_MERGED                       ; 1315
                                 03  12 000C0          BNEQ    13$
                     52          62  D0 000C2          MOVL    (RDB), RDB                              ; 1316
                10   A3          52  D0 000C5 13$:     MOVL    RDB, 16(FDB)                            ; 1318
                     53     F4   A9  9E 000C9          MOVAB   DIF$GL_REVFDB, FDB                      ; 1320
                     50          67  9A 000CD          MOVZBL  STARS, R0                               ; 1321
                     50          02  C6 000D0          DIVL2   #2, R0
                04   AE          50  B0 000D3          MOVW    R0, STARDESC
          FF74       56          01  F1 000D7          ACBL    #2, #1, I, 3$                           ; 1278
                04   AE          67  9B 000DD          MOVZBW  STARS, STARDESC                         ; 1324
                           04   AE  9F  000E1          PUSHAB  STARDESC                                ; 1325
                     6A          01  FB 000E4          CALLS   #1, PUT_DESC
                     03          55  E8 000E7          BLBS    DONE, 14$                               ; 1328
                                 FF5D 31 000EA          BRW     2$
                     50          01  D0 000ED 14$:     MOVL    #1, R0                                  ; 1330
                                 04  000F0            RET                                             ; 1331

; Routine Size:  241 bytes,    Routine Base:  $CODE$ + 078B
```

```
;  1040     1332  1 ROUTINE output_parallel =
;  1041     1333  2 BEGIN
;  1042     1334  2
;  1043     1335  2 !++
;  1044     1336  2 !
;  1045     1337  2 ! FUNCTIONAL DESCRIPTION:
;  1046     1338  2 !
;  1047     1339  2 !       This routine is called to output the most recent set of detected
;  1048     1340  2 !       differences in PARALLEL format.  Note that unlike all the other output
;  1049     1341  2 !       routines, except SLP, this routine assumes it is being called on the
;  1050     1342  2 !       fly as difference sections are being discovered.
;  1051     1343  2 !
;  1052     1344  2 ! IMPLICIT INPUTS:
;  1053     1345  2 !
;  1054     1346  2 !       The FDB's of the master and revision files.
;  1055     1347  2 !
;  1056     1348  2 ! OUTPUTS:
;  1057     1349  2 !
;  1058     1350  2 !       The differences are written to the output file in parallel format.
;  1059     1351  2 !
;  1060     1352  2 ! ROUTINE VALUES:
;  1061     1353  2 !
;  1062     1354  2 !       Always true
;  1063     1355  2 !
;  1064     1356  2 !--
;  1065     1357  2
;  1066     1358  2 LOCAL
;  1067     1359  2     linedesc : BBLOCK [dsc$c_s_bln],        ! Descriptor for output line
;  1068     1360  2     match,                                  ! Number of match records output to date
;  1069     1361  2     masrdb : REF BBLOCK,                    ! Address of RDB of current record from master file
;  1070     1362  2     revrdb : REF BBLOCK;                    ! Address of RDB of current record from revision file
;  1071     1363  2
;  1072     1364  2 IF .dif$gl_flags [dif$v_init]               ! If init flag is set
;  1073     1365  3    THEN BEGIN                              ! Then output listing header
;  1074     1366  3         dif$gl_flags [dif$v_init] = false; ! And reset flag
;  1075     1367  3         put_parallel_idline ();
;  1076     1368  2         END;
;  1077     1369  2
;  1078     1370  2 masrdb = .dif$gl_masfdb [fdb$l_firstdif];  ! Get first difference record from each file
;  1079     1371  2 revrdb = .dif$gl_revfdb [fdb$l_firstdif];
;  1080     1372  2
;  1081     1373  2 !
;  1082     1374  2 ! Initialize the output descriptor and fill it with dashes.
;  1083     1375  2 !
;  1084     1376  2 linedesc [dsc$w_length] = .dif$gl_parwidth;
;  1085     1377  2 linedesc [dsc$a_pointer] = .dif$gl_outbuf;
;  1086     1378  2 CH$FILL (%ASCII '-', .linedesc [dsc$w_length], .linedesc [dsc$a_pointer]);
;  1087     1379  2
;  1088     1380  2 !
;  1089     1381  2 ! If line numbers are requested, then insert them amidst the dashes.
;  1090     1382  2 ! Either way, output the line of dashes.
;  1091     1383  2 !
;  1092     1384  2 IF .dif$gl_flags [dif$v_linenum]
;  1093     1385  3    THEN BEGIN
;  1094     1386  3         linedesc [dsc$w_length] = .dif$gl_parwidth/ 4;
;  1095     1387  3         insert_linenum (.masrdb, linedesc, 1);
;  1096     1388  3         linedesc [dsc$w_length] = 3 * .dif$gl_parwidth/ 4;
```

```
;  1097     1389  3          insert_linenum (.revrdb, linedesc, 1);
;  1098     1390  3          linedesc [dsc$w_length] = .dif$gl_parwidth;
;  1099     1391  3          END;
;  1100     1392  2      put_desc (linedesc);
;  1101     1393  2
;  1102     1394  2      !
;  1103     1395  2      ! While the difference sections are of equal length, output the difference
;  1104     1396  2      ! records with text from both files.
;  1105     1397  2      !
;  1106     1398  2      WHILE (NOT .masrdb [rdb$v_matchone] AND NOT .revrdb [rdb$v_matchone]
;  1107     1399  2        AND NOT .masrdb [rdb$v_eof] AND NOT .revrdb [rdb$v_eof])
;  1108     1400  2      DO IF .masrdb [rdb$v_ignored]
;  1109     1401  2           THEN masrdb = .masrdb [rdb$l_flink]
;  1110     1402  2           ELSE IF .revrdb [rdb$v_ignored]
;  1111     1403  2                THEN revrdb = .revrdb [rdb$l_flink]
;  1112     1404  2                ELSE BEGIN
;  1113     1405  2                     put_record_parallel (.masrdb, .revrdb);
;  1114     1406  2                     masrdb = .masrdb [rdb$l_flink];
;  1115     1407  2                     revrdb = .revrdb [rdb$l_flink];
;  1116     1408  2                     END;
;  1117     1409  2
;  1118     1410  2      !
;  1119     1411  2      ! While the master file difference section is longer than the revision
;  1120     1412  2      ! file difference section, output the difference records with text from
;  1121     1413  2      ! only the master file.
;  1122     1414  2      !
;  1123     1415  2      WHILE (NOT .masrdb [rdb$v_matchone] AND NOT .masrdb [rdb$v_eof])
;  1124     1416  2      DO BEGIN
;  1125     1417  2         IF NOT .masrdb [rdb$v_ignored]
;  1126     1418  2            THEN put_record_parallel (.masrdb, 0);
;  1127     1419  2         masrdb = .masrdb [rdb$l_flink];
;  1128     1420  2         END;
;  1129     1421  2
;  1130     1422  2      !
;  1131     1423  2      ! While the revision file difference section is longer than the master
;  1132     1424  2      ! file difference section, output the difference records with text from
;  1133     1425  2      ! only the revision file.
;  1134     1426  2      !
;  1135     1427  3      WHILE (NOT .revrdb [rdb$v_matchone] AND NOT .revrdb [rdb$v_eof])
;  1136     1428  3      DO BEGIN
;  1137     1429  3         IF NOT .revrdb [rdb$v_ignored]
;  1138     1430  3            THEN put_record_parallel (0, .revrdb);
;  1139     1431  3         revrdb = .revrdb [rdb$l_flink];
;  1140     1432  2         END;
;  1141     1433  2
;  1142     1434  2      !
;  1143     1435  2      ! Output matches from both files, until we are either done, or one of the
;  1144     1436  2      ! files runs out of records.
;  1145     1437  2      !
;  1146     1438  2      match = 0;
;  1147     1439  3      WHILE (.match NEQU .dif$gl_parallel) AND (NOT .masrdb [rdb$v_eof])
;  1148     1440  3            AND (NOT .revrdb [rdb$v_eof])
;  1149     1441  2      DO IF .masrdb [rdb$v_ignored]
;  1150     1442  2           THEN masrdb = .masrdb [rdb$l_flink]
;  1151     1443  2           ELSE IF .revrdb [rdb$v_ignored]
;  1152     1444  2                THEN revrdb = .revrdb [rdb$l_flink]
;  1153     1445  3                ELSE BEGIN
```

```
;  1154          1446  3                    put_record_parallel (.masrdb, .revrdb);
;  1155          1447                       masrdb = .masrdb [rdb$l_flink];
;  1156          1448                       revrdb = .revrdb [rdb$l_flink];
;  1157          1449                       match = .match + 1;
;  1158          1450                       END;
;  1159          1451
;  1160          1452  2 RETURN true;
;  1161          1453  1 END;                                          ! Of output_parallel


                                        0FFC 00000 OUTPUT_PARALLEL:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1332
                                5B 00000000G   00   9E 00002         MOVAB   DIF$GL_FLAGS, R11
                                5A 00000000G   00   9E 00009         MOVAB   DIF$GL_PARWIDTH, R10
                                59 00000000V   EF   9E 00010         MOVAB   PUT_RECORD_PARALLEL, R9
                                5E             08   C2 00017         SUBL2   #8, SP
                          0B        01     AB         05   E1 0001A   BBC     #5, DIF$GL_FLAGS+1, 1$          ; 1364
                            01      AB               20   8A 0001F   BICB2   #32, DIF$GL_FLAGS+1             ; 1366
                    00000000V       EF               00   FB 00023   CALLS   #0, PUT_PARALLEL_IDLINE        ; 1367
                                57 00000000G   00   D0 0002A 1$:     MOVL    DIF$GL_MASFDB+12, MASRDB        ; 1370
                                56 00000000G   00   D0 00031         MOVL    DIF$GL_REVFDB+12, REVRDB        ; 1371
                                58             6A   D0 00038         MOVL    DIF$GL_PARWIDTH, R8             ; 1376
                                6E             58   B0 0003B         MOVW    R8, LINEDESC
                            04  AE 00000000G   00   D0 0003E         MOVL    DIF$GL_OUTBUF, LINEDESC+4       ; 1377
                6E          2D                 6E   00   2C 00046     MOVC5   #0, (SP), #45, LINEDESC, @LINEDESC+4 ; 1378
                                                     04   BE 0004B
                31        01      AB               04   E1 0004D   BBC     #4, DIF$GL_FLAGS+1, 2$          ; 1384
                50                 58               04   C7 00052   DIVL3   #4, R8, R0                     ; 1386
                                   6E               50   B0 00056   MOVW    R0, LINEDESC
                                                    01   DD 00059   PUSHL   #1                             ; 1387
                                                04  AE   9F 0005B   PUSHAB  LINEDESC
                                                    57   DD 0005E   PUSHL   MASRDB
                    00000000V       EF               03   FB 00060   CALLS   #3, INSERT_LINENUM
                50                 6A               03   C5 00067   MULL3   #3, DIF$GL_PARWIDTH, R0       ; 1388
                51                 50               04   C7 0006B   DIVL3   #4, R0, R1
                                   6E               51   B0 0006F   MOVW    R1, LINEDESC
                                                    01   DD 00072   PUSHL   #1                             ; 1389
                                                04  AE   9F 00074   PUSHAB  LINEDESC
                                                    56   DD 00077   PUSHL   REVRDB
                    00000000V       EF               03   FB 00079   CALLS   #3, INSERT_LINENUM
                                    6E               6A   B0 00080   MOVW    DIF$GL_PARWIDTH, LINEDESC       ; 1390
                                                     5E   DD 00083 2$:  PUSHL   SP                          ; 1392
                    00000000V       EF               01   FB 00085   CALLS   #1, PUT_DESC
                45          08      A7               05   E0 0008C 3$:  BBS     #5, 8(MASRDB), 8$           ; 1398
                26          08      A6               05   E0 00091   BBS     #5, 8(REVRDB), 6$
                21          08      A7               02   E0 00096   BBS     #2, 8(MASRDB), 6$           ; 1399
                1C          08      A6               02   E0 0009B   BBS     #2, 8(REVRDB), 6$
                                    05           08  A7   E9 000A0   BLBC    8(MASRDB), 4$                   ; 1400
                                    57               67   D0 000A4   MOVL    (MASRDB), MASRDB                ; 1401
                                                     E3   11 000A7   BRB     3$
                                    0A           08  A6   E8 000A9 4$:  BLBS    8(REVRDB), 5$               ; 1402
                                                     56   DD 000AD   PUSHL   REVRDB                         ; 1405
                                                     57   DD 000AF   PUSHL   MASRDB
                                    69               02   FB 000B1   CALLS   #2, PUT_RECORD_PARALLEL
```

```
                                57              67  D0 000B4             MOVL    (MASRDB), MASRDB                        ; 1406
                                56              66  D0 000B7 5$:        MOVL    (REVRDB), REVRDB                        ; 1407
                                                D0  11 000BA             BRB     3$                                      ; 1400
                15      08      A7              05  E0 000BC 6$:        BBS     #5, 8(MASRDB), 8$                       ; 1415
                10      08      A7              02  E0 000C1             BBS     #2, 8(MASRDB), 8$
                                07      08      A7  E8 000C6             BLBS    8(MASRDB), 7$                           ; 1417
                                                7E  D4 000CA             CLRL    -(SP)                                   ; 1418
                                                57  DD 000CC             PUSHL   MASRDB
                                69              02  FB 000CE             CALLS   #2, PUT_RECORD_PARALLEL
                                57              67  D0 000D1 7$:        MOVL    (MASRDB), MASRDB                        ; 1419
                                                E6  11 000D4             BRB     6$                                      ; 1415
                15      08      A6              05  E0 000D6 8$:        BBS     #5, 8(REVRDB), 10$                      ; 1427
                10      08      A6              02  E0 000DB             BBS     #2, 8(REVRDB), 10$
                                07      08      A6  E8 000E0             BLBS    8(REVRDB), 9$                           ; 1429
                                                56  DD 000E4             PUSHL   REVRDB                                  ; 1430
                                                7E  D4 000E6             CLRL    -(SP)
                                69              02  FB 000E8             CALLS   #2, PUT_RECORD_PARALLEL
                                56              66  D0 000EB 9$:        MOVL    (REVRDB), REVRDB                        ; 1431
                                                E6  11 000EE             BRB     8$                                      ; 1427
                                                52  D4 000F0 10$:        CLRL    MATCH                                   ; 1438
            00000000G   00                      52  D1 000F2 11$:        CMPL    MATCH, DIF$GL_PARALLEL                  ; 1439
                                                2D  13 000F9             BEQL    14$
                28      08      A7              02  E0 000FB             BBS     #2, 8(MASRDB), 14$
                23      08      A6              02  E0 00100             BBS     #2, 8(REVRDB), 14$                      ; 1440
                                05      08      A7  E9 00105             BLBC    8(MASRDB), 12$                          ; 1441
                                57              67  D0 00109             MOVL    (MASRDB), MASRDB                        ; 1442
                                                E4  11 0010C             BRB     11$                                     ; 1443
                                05      08      A6  E9 0010E 12$:        BLBC    8(REVRDB), 13$                          ; 1444
                                56              66  D0 00112             MOVL    (REVRDB), REVRDB
                                                DB  11 00115             BRB     11$                                     ; 1446
                                                56  DD 00117 13$:        PUSHL   REVRDB
                                                57  DD 00119             PUSHL   MASRDB
                                69              02  FB 0011B             CALLS   #2, PUT_RECORD_PARALLEL                 ; 1447
                                57              67  D0 0011E             MOVL    (MASRDB), MASRDB                        ; 1448
                                56              66  D0 00121             MOVL    (REVRDB), REVRDB                        ; 1449
                                                52  D6 00124             INCL    MATCH                                   ; 1441
                                                CA  11 00126             BRB     11$                                     ; 1452
                                50              01  D0 00128 14$:        MOVL    #1, R0                                  ; 1453
                                                    04 0012B             RET

; Routine Size:  300 bytes,    Routine Base:  $CODE$ + 087C
```

```
;  1163          1454  1 ROUTINE output_separated (fdb)  =
;  1164          1455  2 BEGIN
;  1165          1456  2
;  1166          1457  2 !++
;  1167          1458  2 !
;  1168          1459  2 ! FUNCTIONAL DESCRIPTION:
;  1169          1460  2 !
;  1170          1461  2 !       This routine is called to output the most recent set of detected
;  1171          1462  2 !       differences from a particular input file in SEPARATED format.
;  1172          1463  2 !
;  1173          1464  2 ! INPUTS:
;  1174          1465  2 !
;  1175          1466  2 !       fdb = The address of the FDB of the desired input file.
;  1176          1467  2 !
;  1177          1468  2 ! OUTPUTS:
;  1178          1469  2 !
;  1179          1470  2 !       The differences are written to the separated output file.
;  1180          1471  2 !
;  1181          1472  2 ! ROUTINE VALUES:
;  1182          1473  2 !
;  1183          1474  2 !       Always true
;  1184          1475  2 !
;  1185          1476  2 !--
;  1186          1477  2
;  1187          1478  2 MAP
;  1188          1479  2     fdb : REF BBLOCK;
;  1189          1480  2
;  1190          1481  2 LOCAL
;  1191          1482  2     rdb : REF BBLOCK,                          ! Address of the RDB of the current record
;  1192          1483  2     stardesc : BBLOCK [dsc$c_s_bln];           ! Descriptor for stars
;  1193          1484  2
;  1194          1485  2 IF .dif$gl_flags [dif$v_init]                  ! If init flag is set
;  1195          1486  3    THEN BEGIN                                  ! Then output listing header
;  1196          1487  3         dif$gl_flags [dif$v_init] = false;     ! And reset flag
;  1197          1488  3         stardesc [dsc$w_length] = .stars [0];
;  1198          1489  3         stardesc [dsc$a_pointer] = stars [1];
;  1199          1490  3         put_desc (stardesc);
;  1200          1491  3         put_idline (.fdb);
;  1201          1492  2         END;
;  1202          1493  2
;  1203          1494  2 rdb = .fdb [fdb$l_firstdif];                  ! Get first difference record
;  1204          1495  2
;  1205          1496  3 WHILE (.rdb NEQ .fdb [fdb$l_compnrec])        ! Output all unmatched records
;  1206          1497  3 DO BEGIN
;  1207          1498  3    IF NOT .rdb [rdb$v_match]                  ! If difference
;  1208          1499  4       THEN BEGIN                              ! Then, output it
;  1209          1500  4            fdb [fdb$l_currec] = .rdb;         ! Specify output record
;  1210          1501  4            put_record (.fdb, 0);              ! Output the record
;  1211          1502  3            END;
;  1212          1503  3    rdb = .rdb [rdb$l_flink];                  ! Get the next record
;  1213          1504  2    END;
;  1214          1505  2
;  1215          1506  2 RETURN true;
;  1216          1507  1 END;
```

```
                                000C 00000 OUTPUT_SEPARATED:
                                           .WORD   Save R2,R3                                      ; 1454
                          5E            08  C2 00002          SUBL2   #8, SP                      ;
                29 00000000G    00      05  E1 00005          BBC     #5, DIF$GL_FLAGS+1, 1$      ; 1485
                   00000000G    00      20  8A 0000D          BICB2   #32, DIF$GL_FLAGS+1         ; 1487
                          6E 00000000'  EF  9B 00014          MOVZBW  STARS, STARDESC             ; 1488
                       04 AE 00000000'  EF  9E 0001B          MOVAB   STARS+1, STARDESC+4         ; 1489
                                        5E  DD 00023          PUSHL   SP                          ; 1490
                   00000000V    EF      01  FB 00025          CALLS   #1, PUT_DESC                ;
                                    04  AC  DD 0002C          PUSHL   FDB                         ; 1491
                   00000000V    EF      01  FB 0002F          CALLS   #1, PUT_IDLINE              ;
                          52    04      AC  D0 00036 1$:      MOVL    FDB, R2                     ; 1494
                          53    0C      A2  D0 0003A          MOVL    12(R2), RDB                 ;
                       14 A2            53  D1 0003E 2$:      CMPL    RDB, 20(R2)                 ; 1496
                                        18  13 00042          BEQL    4$                          ;
                0E     08 A3            04  E0 00044          BBS     #4, 8(RDB), 3$              ; 1498
                          62            53  D0 00049          MOVL    RDB, (R2)                   ; 1500
                                        7E  D4 0004C          CLRL    -(SP)                       ; 1501
                                        52  DD 0004E          PUSHL   R2                          ;
                   00000000V    EF      02  FB 00050          CALLS   #2, PUT_RECORD              ;
                          53            63  D0 00057 3$:      MOVL    (RDB), RDB                  ; 1503
                                        E2  11 0005A          BRB     2$                          ; 1496
                          50            01  D0 0005C 4$:      MOVL    #1, R0                      ; 1506
                                            04 0005F          RET                                 ; 1507

; Routine Size:  96 bytes,    Routine Base:  $CODE$ + 09A8
```

```
;  1218          1508  1 ROUTINE output_slp =
;  1219          1509  2 BEGIN
;  1220          1510  2
;  1221          1511  2 !++
;  1222          1512  2 !
;  1223          1513  2 ! FUNCTIONAL DESCRIPTION:
;  1224          1514  2 !
;  1225          1515  2 !       This routine is called to output the most recent edit to the
;  1226          1516  2 !       SLP output file.  Unlike all the other output routine, except
;  1227          1517  2 !       PARALLEL, this routine assumes that it is being called on the
;  1228          1518  2 !       fly, as each new difference section is detected.
;  1229          1519  2 !
;  1230          1520  2 ! IMPLICIT INPUTS:
;  1231          1521  2 !
;  1232          1522  2 !       The FDB's of the two input files.
;  1233          1523  2 !
;  1234          1524  2 ! OUTPUTS:
;  1235          1525  2 !
;  1236          1526  2 !       The edits are written to the output file in SLP format.
;  1237          1527  2 !
;  1238          1528  2 ! ROUTINE VALUES:
;  1239          1529  2 !
;  1240          1530  2 !       Always true
;  1241          1531  2 !
;  1242          1532  2 !--
;  1243          1533  2
;  1244          1534  2 LOCAL
;  1245          1535  2     charptr,                                   ! Pointer into output buffer
;  1246          1536  2     linedesc : BBLOCK [dsc$c_s_bln],           ! Descriptor of output line
;  1247          1537  2     number,                                    ! Record number
;  1248          1538  2     rdb : REF BBLOCK;                          ! Address of the RDB of the current record
;  1249          1539  2
;  1250          1540  2 BIND
;  1251          1541  2     firstrdb = .dif$gl_masfdb [fdb$l_firstdif] : BBLOCK,       ! RDB of first record in difference section
;  1252          1542  2     compnrdb = .dif$gl_masfdb [fdb$l_compnrec] : BBLOCK;       ! RDB of first match after difference sectio
;  1253          1543  2
;  1254          1544  2
;  1255          1545  2 IF .dif$gl_flags [dif$v_init]                  ! If init flag is set
;  1256          1546  2    THEN dif$gl_flags [dif$v_init] = false;      ! Then reset flag
;  1257          1547  2
;  1258          1548  2 charptr = .dif$gl_outbuf;                     ! Init ptr to output buffer
;  1259          1549  2 CH$WCHAR_A (%C'-', charptr);                   ! Insert "-" in output buffer
;  1260          1550  2 linedesc [dsc$b_class] = dsc$k_class_s;        ! Set desc class for RTL routine
;  1261          1551  2
;  1262          1552  2 IF firstrdb EQL compnrdb                      ! Are we deleting any lines?
;  1263          1553  2    THEN number = .firstrdb [rdb$l_number]       ! No, then get starting point of insertion
;  1264          1554  3    ELSE BEGIN                                   ! Yes, then we must be replacing lines
;  1265          1555  3         linedesc [dsc$w_length] = dif$c_linenum - 1;    ! Put number of first line to replace into
;  1266          1556  3         linedesc [dsc$a_pointer] = .charptr;            ! the output buffer
;  1267          1557  3         OTS$CVT_L_TI (firstrdb [rdb$l_number], linedesc);
;  1268          1558  3         charptr = .charptr + dif$c_linenum - 1;
;  1269          1559  3         number = .compnrdb [rdb$l_number];      ! Get number of line after last line to replace
;  1270          1560  2         END;
;  1271          1561  2
;  1272          1562  2 IF (number = .number - 1) NEQ 0                ! Calculate number of last line to replace
;  1273          1563  2                                               ! or point of insertion
;  1274          1564  3    THEN BEGIN                                  ! If not at beginning of file
```

```
;  1275     1565  3          IF firstrdb NEQ compnrdb                          ! Then if replacing,
;  1276     1566  3             THEN CH$WCHAR_A (%C',', charptr);              ! Then insert a comma in the output buffer
;  1277     1567  3          linedesc [dsc$w_length] = dif$c_linenum - 1;       ! Insert the number in the output buffer
;  1278     1568  3          linedesc [dsc$a_pointer] = .charptr;
;  1279     1569  3          OTS$CVT_L_TI (number, linedesc);
;  1280     1570  3          charptr = .charptr + dif$c_linenum - 1;
;  1281     1571  2          END;
;  1282     1572  2
;  1283     1573  2    linedesc [dsc$a_pointer] = .dif$gl_outbuf;               ! Initialize the output buffer
;  1284     1574  2    linedesc [dsc$w_length] = .charptr - .dif$gl_outbuf;     ! Set length to amount of buffer already used
;  1285     1575  2    put_desc (linedesc);                                     ! Output the edit command
;  1286     1576  2
;  1287     1577  2    rdb = .dif$gl_revfdb [fdb$l_firstdif];                   ! Get first record of insertion
;  1288     1578  2
;  1289     1579  3    WHILE (.rdb NEQ .dif$gl_revfdb [fdb$l_compnrec])         ! Output each record of the insertion
;  1290     1580  3    DO BEGIN
;  1291     1581  3       dif$gl_revfdb [fdb$l_currec] = .rdb;                  ! Specify the output record
;  1292     1582  3       put_record (dif$gl_revfdb, 0);                        ! Output the record
;  1293     1583  3       rdb = .rdb [rdb$l_flink];                             ! Get the next record
;  1294     1584  2       END;
;  1295     1585  2
;  1296     1586  2    RETURN true;
;  1297     1587  1    END;


                                00FC 00000 OUTPUT_SLP:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1508
                    57 00000000G   00  9E 00002    MOVAB   OTS$CVT_L_TI, R7                ;
                    56 00000000G   00  9E 00009    MOVAB   DIF$GL_OUTBUF, R6               ;
                    55 00000000G   00  9E 00010    MOVAB   DIF$GL_REVFDB, R5               ;
                    5E             0C  C2 00017    SUBL2   #12, SP                         ;
                    52 00000000G   00  D0 0001A    MOVL    DIF$GL_MASFDB+12, R2            ; 1541
                    53 00000000G   00  D0 00021    MOVL    DIF$GL_MASFDB+20, R3            ; 1542
          07 00000000G   00            05  E1 00028    BBC     #5, DIF$GL_FLAGS+1, 1$          ; 1545
             00000000G   00            20  8A 00030    BICB2   #32, DIF$GL_FLAGS+1             ; 1546
                    54                 66  D0 00037 1$:  MOVL    DIF$GL_OUTBUF, CHARPTR          ; 1548
                    84                 2D  90 0003A    MOVB    #45, (CHARPTR)+                 ; 1549
                07  AE                 01  90 0003D    MOVB    #1, LINEDESC+3                  ; 1550
                    53                 52  D1 00041    CMPL    R2, R3                          ; 1552
                                       06  12 00044    BNEQ    2$                              ;
                    6E             04  A2  D0 00046    MOVL    4(R2), NUMBER                   ; 1553
                                       18  11 0004A    BRB     3$                              ;
                04  AE                 05  B0 0004C 2$:  MOVW    #5, LINEDESC                    ; 1555
                08  AE                 54  D0 00050    MOVL    CHARPTR, LINEDESC+4             ; 1556
                                   04  AE  9F 00054    PUSHAB  LINEDESC                        ; 1557
                                   04  A2  9F 00057    PUSHAB  4(R2)                           ;
                    67                 02  FB 0005A    CALLS   #2, OTS$CVT_L_TI                ;
                    54                 05  C0 0005D    ADDL2   #5, CHARPTR                     ; 1558
                    6E             04  A3  D0 00060    MOVL    4(R3), NUMBER                   ; 1559
                                       6E  D7 00064 3$:  DECL    NUMBER                          ; 1562
                                       1C  13 00066    BEQL    5$                              ;
                    53                 52  D1 00068    CMPL    R2, R3                          ; 1565
                                       03  13 0006B    BEQL    4$                              ;
                    84                 2C  90 0006D    MOVB    #44, (CHARPTR)+                 ; 1566
```

```
                04   AE              05   B0  00070 4$:    MOVW    #5, LINEDESC                            ; 1567
                08   AE              54   D0  00074        MOVL    CHARPTR, LINEDESC+4                     ; 1568
                                 04   AE   9F  00078        PUSHAB  LINEDESC                                ; 1569
                                 04   AE   9F  0007B        PUSHAB  NUMBER                                  ;
                     67                  02   FB  0007E    CALLS   #2, OTS$CVT_L_TI                        ;
                     54                  05   C0  00081    ADDL2   #5, CHARPTR                             ; 1570
                     50                  66   D0  00084 5$: MOVL    DIF$GL_OUTBUF, R0                       ; 1573
                08   AE                  50   D0  00087    MOVL    R0, LINEDESC+4                          ;
          04    AE                       54   A3  0008B    SUBW3   R0, CHARPTR, LINEDESC                   ; 1574
                                 04   AE   9F  00090        PUSHAB  LINEDESC                                ; 1575
           00000000V             EF   01   FB  00093        CALLS   #1, PUT_DESC                            ;
                     52          0C   A5   D0  0009A        MOVL    DIF$GL_REVFDB+12, RDB                   ; 1577
                14   A5             52   D1  0009E 6$:     CMPL    RDB, DIF$GL_REVFDB+20                   ; 1579
                                        13   13  000A2        BEQL    7$                                      ;
                     65                  52   D0  000A4        MOVL    RDB, DIF$GL_REVFDB                      ; 1581
                                         7E   D4  000A7        CLRL    -(SP)                                   ; 1582
                                         55   DD  000A9        PUSHL   R5                                      ;
           00000000V             EF        02   FB  000AB    CALLS   #2, PUT_RECORD                          ;
                     52                  62   D0  000B2        MOVL    (RDB), RDB                              ; 1583
                                         E7   11  000B5        BRB     6$                                      ; 1579
                     50                  01   D0  000B7 7$:     MOVL    #1, R0                                  ; 1586
                                             04   000BA        RET                                             ; 1587

; Routine Size:  187 bytes,    Routine Base:  $CODE$ + 0A08
```

```
;  1299     1588  1 GLOBAL ROUTINE put_record (fdb, cbarflag) =
;  1300     1589  2 BEGIN
;  1301     1590  2
;  1302     1591  2 !++
;  1303     1592  2 !
;  1304     1593  2 ! FUNCTIONAL DESCRIPTION:
;  1305     1594  2 !
;  1306     1595  2 !       Call the appropriate radix output routine to format and put a
;  1307     1596  2 !       record to the output file.
;  1308     1597  2 !
;  1309     1598  2 ! INPUTS:
;  1310     1599  2 !
;  1311     1600  2 !       fdb =       The address of the FDB pointing to the CURREC that is
;  1312     1601  2 !                   to be output.
;  1313     1602  2 !
;  1314     1603  2 !       cbarflag =  A flag that is 0 if not changebar format
;  1315     1604  2 !                                  1 if changebar format and bar should be output
;  1316     1605  2 !                                  2 if changebar format and bar should not be ouput
;  1317     1606  2 !
;  1318     1607  2 ! OUTPUTS:
;  1319     1608  2 !
;  1320     1609  2 !       The CURREC is output.
;  1321     1610  2 !
;  1322     1611  2 ! ROUTINE VALUES:
;  1323     1612  2 !
;  1324     1613  2 !       Always true
;  1325     1614  2 !
;  1326     1615  2 !--
;  1327     1616  2 MAP
;  1328     1617  2     fdb : REF BBLOCK;
;  1329     1618  2
;  1330     1619  2 LOCAL
;  1331     1620  2     rdb : REF BBLOCK,
;  1332     1621  2     stardesc : BBLOCK [dsc$c_s_bln];
;  1333     1622  2
;  1334     1623  2 !
;  1335     1624  2 ! If init flag is set, then this must be a change bar listing.  Output header
;  1336     1625  2 ! here, instead of in OUTPUT_CHANGEBAR, because we can never tell if the first
;  1337     1626  2 ! record output will be a match or a difference.
;  1338     1627  2 !
;  1339     1628  2 IF .dif$gl_flags [dif$v_init]                          ! If init flag is set
;  1340     1629  3    THEN BEGIN                                          ! Then output listing header
;  1341     1630  3         dif$gl_flags [dif$v_init] = false;             ! And reset flag
;  1342     1631  3         stardesc [dsc$w_length] = .stars [0];
;  1343     1632  3         stardesc [dsc$a_pointer] = stars [1];
;  1344     1633  3         put_desc (stardesc);
;  1345     1634  3         put_idline (.fdb);
;  1346     1635  3         END;
;  1347     1636  2
;  1348     1637  2 rdb = .fdb [fdb$l_currec];                             ! Get address of RDB of record to output
;  1349     1638  2
;  1350     1639  2 IF .rdb [rdb$v_eof] OR .rdb [rdb$v_ignored]           ! If EOF or ignore
;  1351     1640  2    THEN RETURN true;                                   ! Then don't output, simply return
;  1352     1641  2
;  1353     1642  2 IF .dif$gl_flags [dif$v_ascii]                        ! Call appropriate mode record output routin
;  1354     1643  2    THEN put_record_ascii (.fdb, .cbarflag)
;  1355     1644  2    ELSE put_record_hex_octal (.fdb, .cbarflag);
```

```
;  1356          1645  2
;  1357          1646  2 RETURN true;
;  1358          1647  1 END;


                                                    0004 00000         .ENTRY  PUT_RECORD, Save R2                             ;  1588
                                 52 00000000G   00  9E 00002         MOVAB   DIF$GL_FLAGS, R2                                ;
                                 5E             08  C2 00009         SUBL2   #8, SP                                          ;
                          26      01 A2         05  E1 0000C         BBC     #5, DIF$GL_FLAGS+1, 1$                          ;  1628
                                  01 A2         20  8A 00011         BICB2   #32, DIF$GL_FLAGS+1                             ;  1630
                                 6E 00000000'   EF  9B 00015         MOVZBW  STARS, STARDESC                                 ;  1631
                                  04 AE 00000000'   EF  9E 0001C     MOVAB   STARS+1, STARDESC+4                             ;  1632
                                                 5E  DD 00024         PUSHL   SP                                              ;  1633
                              00000000V  EF      01  FB 00026         CALLS   #1, PUT_DESC                                    ;
                                      04 AC      DD 0002D            PUSHL   FDB                                             ;  1634
                              00000000V  EF      01  FB 00030         CALLS   #1, PUT_IDLINE                                  ;
                                 50      04 BC  D0 00037  1$:         MOVL    @FDB, RDB                                       ;  1637
                          1F      08 A0         02  E0 0003B         BBS     #2, 8(RDB), 3$                                  ;  1639
                                 1B      08 A0  E8 00040             BLBS    8(RDB), 3$                                      ;
                                 0D             62  E9 00044         BLBC    DIF$GL_FLAGS, 2$                                ;  1642
                                 7E      04 AC  7D 00047             MOVQ    FDB, -(SP)                                      ;  1643
                              00000000V  EF      02  FB 0004B         CALLS   #2, PUT_RECORD_ASCII                            ;
                                                 0B  11 00052         BRB     3$                                              ;
                                 7E      04 AC  7D 00054  2$:         MOVQ    FDB, -(SP)                                      ;  1644
                              00000000V  EF      02  FB 00058         CALLS   #2, PUT_RECORD_HEX_OCTAL                        ;
                                 50              01  D0 0005F  3$:    MOVL    #1, R0                                          ;  1646
                                                 04  00062            RET                                                     ;  1647

; Routine Size:  99 bytes,    Routine Base:  $CODE$ + 0AC3
```

```
;  1360          1648  1 ROUTINE put_record_ascii (fdb, cbarflag) =
;  1361          1649  2 BEGIN
;  1362          1650  2
;  1363          1651  2 !++
;  1364          1652  2 !
;  1365          1653  2 ! FUNCTIONAL DESCRIPTION:
;  1366          1654  2 !
;  1367          1655  2 !       Format and put a record to the output file in ascii mode.
;  1368          1656  2 !
;  1369          1657  2 ! INPUTS:
;  1370          1658  2 !
;  1371          1659  2 !       fdb =       The address of the FDB pointing to the CURREC that is
;  1372          1660  2 !                   to be output.
;  1373          1661  2 !
;  1374          1662  2 !       cbarflag =  A flag that is 0 if not changebar format
;  1375          1663  2 !                                  1 if changebar format and bar should be output
;  1376          1664  2 !                                  2 if changebar format and bar should not be ouput
;  1377          1665  2 !
;  1378          1666  2 ! OUTPUTS:
;  1379          1667  2 !
;  1380          1668  2 !       The CURREC is output.
;  1381          1669  2 !
;  1382          1670  2 ! ROUTINE VALUES:
;  1383          1671  2 !
;  1384          1672  2 !       Always true
;  1385          1673  2 !
;  1386          1674  2 !--
;  1387          1675  2
;  1388          1676  2 MAP
;  1389          1677  2     fdb : REF BBLOCK;
;  1390          1678  2
;  1391          1679  2 LOCAL
;  1392          1680  2     charptr,                                  ! Pointer into the output buffer
;  1393          1681  2     linedesc : BBLOCK [dsc$c_s_bln],          ! Descriptor of output line
;  1394          1682  2     rdb : REF BBLOCK,                         ! Address of the RDB of the record to be output
;  1395          1683  2     text_length,                              ! Space in record for text
;  1396          1684  2     status;
;  1397          1685  2
;  1398          1686  2 rdb = .fdb [fdb$l_currec];                                ! Get address of RDB of record to output
;  1399          1687  2
;  1400          1688  2 linedesc [dsc$w_length] = 0;                             ! Init the output descriptor
;  1401          1689  2 linedesc [dsc$a_pointer] = .dif$gl_outbuf;
;  1402          1690  2 charptr = .dif$gl_outbuf;                               ! Init the pointer into the output buffer
;  1403          1691  2
;  1404          1692  2 IF .cbarflag NEQ 0                                      ! If change bar format output
;  1405          1693  2
;  1406          1694  2 !
;  1407          1695  2 ! Change bar format.  If line number is requested, then insert number
;  1408          1696  2 ! into the output buffer.  Then insert change bar or blanks, as
;  1409          1697  2 ! appropriate.
;  1410          1698  2 !
;  1411          1699  3     THEN BEGIN
;  1412          1700  3          IF .fdb [fdb$v_linenum]                       ! If line number should be included
;  1413          1701  4             THEN BEGIN                                 ! Then do so
;  1414          1702  4                  insert_linenum (.rdb, linedesc, 0);   ! Insert line number in the buffer
;  1415          1703  4                  charptr = .charptr + dif$c_linenum;   ! Incr the char ptr
;  1416          1704  3                  END;
```

```
;  1417     1705  3
;  1418     1706  3          IF .cbarflag EQL 1                                  ! If this record requires change bar
;  1419     1707  3             THEN CH$WCHAR_A (.fdb [fdb$b_cbarchr], charptr)  ! Then insert one
;  1420     1708  3             ELSE CH$WCHAR_A (%C' ', charptr);                ! Else leave a space instead
;  1421     1709  3
;  1422     1710  3          CH$WCHAR_A (%C' ', charptr);                        ! Pad with a blank
;  1423     1711  3
;  1424     1712  3          END
;  1425     1713  3
;  1426     1714  !
;  1427     1715  ! If not changebar, then if line numbers requested then insert the line number.
;  1428     1716  !
;  1429     1717     ELSE IF .dif$gl_flags [dif$v_linenum]                       ! If line number should be incuded
;  1430     1718          THEN BEGIN                                             ! Then do so
;  1431     1719               insert_linenum (.rdb, linedesc, 0);               ! Insert line number in the buffer
;  1432     1720               charptr = .charptr + dif$c_linenum;               ! Incr char ptr
;  1433     1721               CH$WCHAR_A (%C' ', charptr);                      ! Pad with two blanks
;  1434     1722               CH$WCHAR_A (%C' ', charptr);
;  1435     1723               END;
;  1436     1724
;  1437     1725  !
;  1438     1726  ! If SLP output, then handle special operators in first column.
;  1439     1727  !
;  1440     1728  IF .dif$gl_flags [dif$v_slp]                                   ! If SLP output
;  1441     1729     THEN IF (.rdb [rdb$w_length] GTR 0) AND                     ! and non-null record
;  1442     1730             (CH$FIND_CH (.slpoprs [0],                          ! and leading operator
;  1443     1731                          slpoprs [1],
;  1444     1732                          CH$RCHAR (rdb [rdb$t_text])) NEQ 0)
;  1445     1733             THEN CH$WCHAR_A (%C'<', charptr);                   ! then insert escape operator
;  1446     1734
;  1447     1735  linedesc [dsc$w_length] = .charptr - .dif$gl_outbuf;           ! Set length to amount of buffer already use
;  1448     1736  IF .dif$gl_width GTRU .linedesc [dsc$w_length]                  ! Proceed only if room remains on line
;  1449     1737  THEN
;  1450     1738      BEGIN
;  1451     1739      IF .dif$gl_ignore [ign$v_exact] AND .rdb [rdb$v_edited]     ! If outputing exact, and if record has been
;  1452     1740      THEN
;  1453     1741          get_rfa_text (.fdb, linedesc, .dif$gl_width)            ! Then get original record using RFA
;  1454     1742      ELSE BEGIN                                                  ! Else use edited string
;  1455     1743          text_length = MINU (.dif$gl_width - .linedesc [dsc$w_length],   ! Calculate amount of space left for
;  1456     1744                              .rdb [rdb$w_length]);
;  1457     1745          CH$MOVE (.text_length, rdb [rdb$t_text], .charptr);     ! Move text into buffer
;  1458     1746          linedesc [dsc$w_length] = .linedesc [dsc$w_length] + .text_length;      ! Update string length
;  1459     1747          END;
;  1460     1748      END;
;  1461     1749
;  1462     1750  IF .dif$gl_ignore [ign$v_pretty]                               ! If PRETTY mode, edit output line
;  1463     1751  THEN
;  1464     1752      linedesc [dsc$w_length] =
;  1465     1753          translate_tabs (.linedesc [dsc$a_pointer], .charptr, .linedesc [dsc$w_length], .dif$gl_width);
;  1466     1754
;  1467     1755  put_desc (linedesc);                                           ! Output the line
;  1468     1756
;  1469     1757  RETURN true;
;  1470     1758  1 END;
```

```
                                OFFC 00000 PUT_RECORD_ASCII:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                    ; 1648
                5B 00000000G   00   9E 00002       MOVAB   DIF$GL_OUTBUF, R11                             ;
                5A 00000000G   00   9E 00009       MOVAB   DIF$GL_IGNORE, R10                             ;
                59 00000000G   00   9E 00010       MOVAB   DIF$GL_WIDTH, R9                               ;
                58 00000000V   EF   9E 00017       MOVAB   INSERT_LINENUM, R8                             ;
                5E             08   C2 0001E       SUBL2   #8, SP                                         ;
                53       04    AC   D0 00021       MOVL    FDB, R3                                        ; 1686
                52             63   D0 00025       MOVL    (R3), RDB                                      ;
                               6E   B4 00028       CLRW    LINEDESC                                       ; 1688
                50             6B   D0 0002A       MOVL    DIF$GL_OUTBUF, R0                              ; 1689
          04    AE             50   D0 0002D       MOVL    R0, LINEDESC+4                                 ;
                56             50   D0 00031       MOVL    R0, CHARPTR                                    ; 1690
                         08    AC   D5 00034       TSTL    CBARFLAG                                       ; 1692
                               1E   13 00037       BEQL    2$                                             ;
    0D    24    A3             01   E1 00039       BBC     #1, 36(R3), 1$                                 ; 1700
                               7E   D4 0003E       CLRL    -(SP)                                          ; 1702
                         04    AE   9F 00040       PUSHAB  LINEDESC                                       ;
                               52   DD 00043       PUSHL   RDB                                            ;
                68             03   FB 00045       CALLS   #3, INSERT_LINENUM                             ;
                56             06   C0 00048       ADDL2   #6, CHARPTR                                    ; 1703
                01       08    AC   D1 0004B 1$:   CMPL    CBARFLAG, #1                                   ; 1706
                               1B   12 0004F       BNEQ    3$                                             ;
                66       26    A3   90 00051       MOVB    38(R3), (CHARPTR)                              ; 1707
                               18   11 00055       BRB     4$                                             ; 1708
    15 00000000G   00          04   E1 00057 2$:   BBC     #4, DIF$GL_FLAGS+1, 5$                         ; 1717
                               7E   D4 0005F       CLRL    -(SP)                                          ; 1719
                         04    AE   9F 00061       PUSHAB  LINEDESC                                       ;
                               52   DD 00064       PUSHL   RDB                                            ;
                68             03   FB 00066       CALLS   #3, INSERT_LINENUM                             ;
                56             06   C0 00069       ADDL2   #6, CHARPTR                                    ; 1720
                66             20   90 0006C 3$:   MOVB    #32, (CHARPTR)                                 ; 1721
                               56   D6 0006F 4$:   INCL    CHARPTR                                        ;
                86             20   90 00071       MOVB    #32, (CHARPTR)+                                ; 1722
                20 00000000G   00   E9 00074 5$:   BLBC    DIF$GL_FLAGS+1, 7$                             ; 1728
                         12    A2   B5 0007B       TSTW    18(RDB)                                        ; 1729
                               1B   13 0007E       BEQL    7$                                             ;
                50 00000000'   EF   9A 00080       MOVZBL  SLPOPRS, R0                                    ; 1730
    00000000'   EF             50   14 A2 00087    LOCC    20(RDB), R0, SLPOPRS+1                         ; 1732
                               02   12 00090       BNEQ    6$                                             ;
                               51   D4 00092       CLRL    R1                                             ;
                               51   D5 00094 6$:   TSTL    R1                                             ;
                               03   13 00096       BEQL    7$                                             ;
                86             3C   90 00098       MOVB    #60, (CHARPTR)+                                ; 1733
          6E    56             6B   A3 0009B 7$:   SUBW3   DIF$GL_OUTBUF, CHARPTR, LINEDESC               ; 1735
                50             69   D0 0009F       MOVL    DIF$GL_WIDTH, R0                               ; 1736
    50    6E    10             00   ED 000A2       CMPZV   #0, #16, LINEDESC, R0                          ;
                               36   1E 000A7       BGEQU   10$                                            ;
          15    6A             06   E1 000A9       BBC     #6, DIF$GL_IGNORE, 8$                          ; 1739
          10    08    A2       03   E1 000AD       BBC     #3, 8(RDB), 8$                                 ;
                               50   DD 000B2       PUSHL   R0                                             ; 1741
                         04    AE   9F 000B4       PUSHAB  LINEDESC                                       ;
                               53   DD 000B7       PUSHL   R3                                             ;
                00000000V      EF   03   FB 000B9  CALLS   #3, GET_RFA_TEXT                               ;
                               1D   11 000C0       BRB     10$                                            ;
```

```
                                 51              6E  3C 000C2 8$:    MOVZWL  LINEDESC, R1                            ; 1743
                                 50              51  C2 000C5       SUBL2   R1, R0
          50      12   A2        10              00  ED 000C8       CMPZV   #0, #16, 18(RDB), R0                    ; 1744
                                                 04  1E 000CE       BGEQU   9$
                                 50      12      A2  3C 000D0       MOVZWL  18(RDB), R0
                                 57              50  D0 000D4 9$:    MOVL    R0, TEXT_LENGTH                         ; 1743
                66       14     A2              57  28 000D7       MOVC3   TEXT_LENGTH, 20(RDB), (CHARPTR)         ; 1745
                                 6E              57  A0 000DC       ADDW2   TEXT_LENGTH, LINEDESC                   ; 1746
                                                 6A  95 000DF 10$:   TSTB    DIF$GL_IGNORE                           ; 1750
                                                 15  18 000E1       BGEQ    11$
                                                 69  DD 000E3       PUSHL   DIF$GL_WIDTH                            ; 1753
                                 7E      04      AE  3C 000E5       MOVZWL  LINEDESC, -(SP)
                                                 56  DD 000E9       PUSHL   CHARPTR
                                         10      AE  DD 000EB       PUSHL   LINEDESC+4
                00000000V        EF              04  FB 000EE       CALLS   #4, TRANSLATE_TABS
                                 6E              50  B0 000F5       MOVW    R0, LINEDESC
                                                 5E  DD 000F8 11$:   PUSHL   SP                                      ; 1755
                00000000V        EF              01  FB 000FA       CALLS   #1, PUT_DESC
                                 50              01  D0 00101       MOVL    #1, R0                                  ; 1757
                                                 04  00104         RET                                             ; 1758

; Routine Size:  261 bytes,    Routine Base:  $CODE$ + 0B26
```

```
;  1472          1759  1 ROUTINE put_record_hex_octal (fdb, cbarflag) =
;  1473          1760  2 BEGIN
;  1474          1761  2
;  1475          1762  2 !++
;  1476          1763  2 !
;  1477          1764  2 ! FUNCTIONAL DESCRIPTION:
;  1478          1765  2 !
;  1479          1766  2 !       Format and put a record to the output file in either hex or octal.
;  1480          1767  2 !
;  1481          1768  2 ! INPUTS:
;  1482          1769  2 !
;  1483          1770  2 !       fdb =       The address of the FDB pointing to the CURREC that is
;  1484          1771  2 !                   to be output.
;  1485          1772  2 !
;  1486          1773  2 !       cbarflag =  A flag that is 0 if not changebar format
;  1487          1774  2 !                                  1 if changebar format and bar should be output
;  1488          1775  2 !                                  2 if changebar format and bar should not be ouput
;  1489          1776  2 !
;  1490          1777  2 ! OUTPUTS:
;  1491          1778  2 !
;  1492          1779  2 !       The CURREC is output.
;  1493          1780  2 !
;  1494          1781  2 ! ROUTINE VALUES:
;  1495          1782  2 !
;  1496          1783  2 !       Always true
;  1497          1784  2 !
;  1498          1785  2 !--
;  1499          1786  2
;  1500          1787  2 MAP
;  1501          1788  2     fdb : REF BBLOCK;
;  1502          1789  2
;  1503          1790  2 LITERAL
;  1504          1791  2     byte_bits = 8,              ! number of bits in a byte
;  1505          1792  2     hex_bits = 4,               ! number of bits in a hexadecimal digit
;  1506          1793  2     oct_bits = 3;               ! number of bits in an octal digit
;  1507          1794  2
;  1508          1795  2 LOCAL
;  1509          1796  2     additional,
;  1510          1797  2     bufferpointer,
;  1511          1798  2     bytenumber,
;  1512          1799  2     bytesperline,
;  1513          1800  2     entsinrec,
;  1514          1801  2     faopointer,
;  1515          1802  2     linedesc : BBLOCK [dsc$c_s_bln],
;  1516          1803  2     outputdesc : BBLOCK [dsc$c_s_bln],
;  1517          1804  2     padbytes,
;  1518          1805  2     tempfaobuf : BBLOCK [dif$c_maxfaosiz],
;  1519          1806  2     tempfaodesc : BBLOCK [dsc$c_s_bln],
;  1520          1807  2     rdb : REF BBLOCK;
;  1521          1808  2
;  1522          1809  2 rdb = .fdb [fdb$l_currec];                          ! Get address of RDB of output record
;  1523          1810  2
;  1524          1811  2 !
;  1525          1812  2 ! Get exact or edited text, as appropriate.
;  1526          1813  2 !
;  1527          1814  2 IF .dif$gl_ignore [ign$v_exact] AND .rdb [rdb$v_edited]
;  1528          1815  3     THEN BEGIN
```

```
;  1529          1816  3          linedesc [dsc$w_length] = 0;
;  1530          1817  3          linedesc [dsc$a_pointer] = .dif$gl_inbuf;
;  1531          1818  3          get_rfa_text (.rdb, linedesc,
;  1532          1819  3                  MAXU (.dif$gl_masrab [rab$w_usz], .dif$gl_revrab [rab$w_usz]));
;  1533          1820  3          END
;  1534          1821  3      ELSE BEGIN
;  1535          1822  3          linedesc [dsc$w_length] = .rdb [rdb$w_length];
;  1536          1823  3          linedesc [dsc$a_pointer] = rdb [rdb$t_text];
;  1537          1824  2          END;
;  1538          1825  2
;  1539          1826  2      !
;  1540          1827  2      ! Output the record header.
;  1541          1828  2      !
;  1542          1829  2      put_blank ();
;  1543          1830  2      put_hex_octal_header (.rdb [rdb$l_number], .linedesc [dsc$w_length], .cbarflag);
;  1544          1831  2      put_blank ();
;  1545          1832  2
;  1546          1833  2      !
;  1547          1834  2      ! Initialize output format parameters.
;  1548          1835  2      !
;  1549          1836  2      bytenumber = 0;
;  1550          1837  2      bytesperline = .dif$gl_entsperline * dif$c_entrysize;
;  1551          1838  2      entsinrec = (.linedesc [dsc$w_length] + dif$c_entrysize - 1) / dif$c_entrysize;
;  1552          1839  2      faopointer = dif$gl_faofulldesc;
;  1553          1840  2
;  1554          1841  2      !
;  1555          1842  2      ! Initialize output descriptor.
;  1556          1843  2      !
;  1557          1844  2      outputdesc [dsc$a_pointer] = .dif$gl_outbuf;
;  1558          1845  2      outputdesc [dsc$w_length] = .dif$gl_dumpwidth;
;  1559          1846  2
;  1560          1847  2      !
;  1561          1848  2      ! Output all the data in the record.
;  1562          1849  2      !
;  1563          1850  2      WHILE .entsinrec GTR 0
;  1564          1851  3      DO BEGIN
;  1565          1852  3
;  1566          1853  3      !
;  1567          1854  3      ! If less than a full line of data left, then prepare to output a partial line.
;  1568          1855  3      !
;  1569          1856  3          IF .linedesc [dsc$w_length] LSSU .bytesperline
;  1570          1857  4          THEN BEGIN
;  1571          1858  4              CH$COPY (.linedesc [dsc$w_length], .linedesc [dsc$a_pointer],        ! Copy partial line, zero fi
;  1572          1859  4                  0, .bytesperline, .dif$gl_inbuf);
;  1573          1860  4              tempfaodesc [dsc$w_length] = dif$c_maxfaosiz;                        ! Set up temporary work area
;  1574          1861  4              tempfaodesc [dsc$a_pointer] = tempfaobuf;
;  1575          1862  4              SYS$FAO (dif$gl_faopartdesc, tempfaodesc, tempfaodesc, .entsinrec);  ! Use FAO to build a fao con
;  1576          1863  4              faopointer = tempfaodesc;                                            ! Use this fao string up ahe
;  1577          1864  4              bufferpointer = .dif$gl_inbuf;                                       ! Use this data
;  1578          1865  4              END
;  1579          1866  3          ELSE bufferpointer = .linedesc [dsc$a_pointer];                          ! If full line, use this dat
;  1580          1867  3
;  1581          1868  3      !
;  1582          1869  3      ! Format the output line.
;  1583          1870  3      !
;  1584          1871  3          dif$format_hex_octal (.bufferpointer, .dif$gl_entsperline, dif$c_entrysize,
;  1585          1872  3                  .bytenumber, .entsinrec, 0, .faopointer, outputdesc);
```

```
;  1586          1873  3
;  1587          1874  3
;  1588          1875  3 !
;  1589          1876  3 ! If partial line, then remove leading zeros and replace with blanks.
;  1590          1877  3 !
;  1591          1878  3     IF .linedesc [dsc$w_length] LSSU .bytesperline
;  1592          1879  4     THEN BEGIN
;  1593          1880  4          padbytes = .dif$gl_dumpwidth - .outputdesc [dsc$w_length];
;  1594          1881  4          !
;  1595          1882  4          ! Calculate number of blanks to proceed last few bytes
;  1596          1883  4          ! The value is :
;  1597          1884  4          !   1 plus length of formatted longword less length of formatted bytes
;  1598          1885  4          !
;  1599          1886  4          IF (additional = .linedesc [dsc$w_length] MOD dif$c_entrysize) GTR 0
;  1600          1887  4             THEN
;  1601          1888  4                IF .dif$gl_flags [dif$v_hex]
;  1602          1889  4                THEN
;  1603          1890  4                   additional = 1
;  1604          1891  4                       + (dif$c_entrysize*byte_bits+(hex_bits-1))/hex_bits
;  1605          1892  4                       - (.additional*byte_bits+(hex_bits-1))/hex_bits
;  1606          1893  4                ELSE
;  1607          1894  4                   additional = 1
;  1608          1895  4                       + (dif$c_entrysize*byte_bits+(oct_bits-1))/oct_bits
;  1609          1896  4                       - (.additional*byte_bits+(oct_bits-1))/oct_bits;
;  1610          1897  4          padbytes = .padbytes + .additional;
;  1611          1898  4          CH$MOVE (.outputdesc [dsc$w_length] - .additional,
;  1612          1899  4                   .outputdesc [dsc$a_pointer] + .additional,
;  1613          1900  4                   .outputdesc [dsc$a_pointer] + .padbytes);
;  1614          1901  4          CH$FILL (%ASCII ' ', .padbytes, .outputdesc [dsc$a_pointer]);
;  1615          1902  4          outputdesc [dsc$w_length] = .dif$gl_dumpwidth;
;  1616          1903  3          END;
;  1617          1904  3
;  1618          1905  3 !
;  1619          1906  3 ! Output the fully formatted line.
;  1620          1907  3 !
;  1621          1908  3     put_desc (outputdesc);
;  1622          1909  3
;  1623          1910  3 !
;  1624          1911  3 ! Update parameters that mark our place in the data.
;  1625          1912  3 !
;  1626          1913  3     entsinrec = .entsinrec - .dif$gl_entsperline;
;  1627          1914  3     bytenumber = .bytenumber + .bytesperline;
;  1628          1915  3     linedesc [dsc$w_length] = .linedesc [dsc$w_length] - .bytesperline;
;  1629          1916  3     linedesc [dsc$a_pointer] = .linedesc [dsc$a_pointer] + .bytesperline;
;  1630          1917  3     END;
;  1631          1918  2
;  1632          1919  2
;  1633          1920  2 RETURN true;
;  1634          1921  1 END;


                                OFFC 00000 PUT_RECORD_HEX_OCTAL:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 1759
                    5E      B8  AE  9E 00002       MOVAB   -72(SP), SP                         ;
```

```
                          52        04  BC  D0 00006       MOVL    @FDB, RDB                                    ; 1809
             36 00000000G   00          06  E1 0000A       BBC     #6, DIF$GL_IGNORE, 2$                        ; 1814
             31        08   A2          03  E1 00012       BBC     #3, 8(RDB), 2$                               ;
                                   40   AE  B4 00017       CLRW    LINEDESC                                     ; 1816
                   44   AE 00000000G    00  D0 0001A       MOVL    DIF$GL_INBUF, LINEDESC+4                     ; 1817
                        7E 00000000G    00  3C 00022       MOVZWL  DIF$GL_MASRAB+32, -(SP)                      ; 1819
                        6E 00000000G    00  B1 00029       CMPW    DIF$GL_REVRAB+32, (SP)                       ;
                                        07  1B 00030       BLEQU   1$                                           ;
                        6E 00000000G    00  3C 00032       MOVZWL  DIF$GL_REVRAB+32, (SP)                       ;
                                   44   AE  9F 00039  1$:  PUSHAB  LINEDESC                                     ; 1818
                                   04   AC  DD 0003C       PUSHL   FDB                                          ;
                00000000V   EF          03  FB 0003F       CALLS   #3, GET_RFA_TEXT                             ;
                                        0A  11 00046       BRB     3$                                           ; 1814
                   40   AE         12   A2  B0 00048  2$:  MOVW    18(RDB), LINEDESC                            ; 1822
                   44   AE         14   A2  9E 0004D       MOVAB   20(R2), LINEDESC+4                           ; 1823
                00000000V   EF          00  FB 00052  3$:  CALLS   #0, PUT_BLANK                                ; 1829
                                   08   AC  DD 00059       PUSHL   CBARFLAG                                     ; 1830
                        7E         44   AE  3C 0005C       MOVZWL  LINEDESC, -(SP)                              ;
                                   04   A2  DD 00060       PUSHL   4(RDB)                                       ;
                00000000V   EF          03  FB 00063       CALLS   #3, PUT_HEX_OCTAL_HEADER                     ;
                00000000V   EF          00  FB 0006A       CALLS   #0, PUT_BLANK                                ; 1831
                                        6E  D4 00071       CLRL    BYTENUMBER                                   ; 1836
             59 00000000G   00          02  78 00073       ASHL    #2, DIF$GL_ENTSPERLINE, BYTESPERLINE         ; 1837
                            50     40   AE  3C 0007B       MOVZWL  LINEDESC, R0                                 ; 1838
                            50          03  C0 0007F       ADDL2   #3, R0                                       ;
             5A             50          04  C7 00082       DIVL3   #4, R0, ENTSINREC                            ;
                        5B 00000000G    00  9E 00086       MOVAB   DIF$GL_FAOFULLDESC, FAOPOINTER               ; 1839
                   3C   AE 00000000G    00  D0 0008D       MOVL    DIF$GL_OUTBUF, OUTPUTDESC+4                  ; 1844
                   38   AE 00000000G    00  B0 00095       MOVW    DIF$GL_DUMPWIDTH, OUTPUTDESC                 ; 1845
                                        5A  D5 0009D  4$:  TSTL    ENTSINREC                                    ; 1850
                                        03  14 0009F       BGTR    5$                                           ;
                                      00EC  31 000A1       BRW     11$                                          ;
                                        58  D4 000A4  5$:  CLRL    R8                                           ; 1856
 59       40   AE          10           00  ED 000A6       CMPZV   #0, #16, LINEDESC, BYTESPERLINE              ;
                                        3D  1E 000AC       BGEQU   6$                                           ;
                                        58  D6 000AE       INCL    R8                                           ;
                        50 00000000G    00  D0 000B0       MOVL    DIF$GL_INBUF, R0                             ; 1859
 59            00      44   BE     40   AE  2C 000B7       MOVC5   LINEDESC, @LINEDESC+4, #0, BYTESPERLINE, -   ;
                                        60     000BE               (R0)                                         ;
                   08   AE              28  B0 000BF       MOVW    #40, TEMPFAODESC                             ; 1860
                   0C   AE         10   AE  9E 000C3       MOVAB   TEMPFAOBUF, TEMPFAODESC+4                    ; 1861
                                        5A  DD 000C8       PUSHL   ENTSINREC                                    ; 1862
                                   0C   AE  9F 000CA       PUSHAB  TEMPFAODESC                                  ;
                                   10   AE  9F 000CD       PUSHAB  TEMPFAODESC                                  ;
                           00000000G    00  9F 000D0       PUSHAB  DIF$GL_FAOPARTDESC                           ;
                00000000G   00          04  FB 000D6       CALLS   #4, SYS$FAO                                  ;
                        5B         08   AE  9E 000DD       MOVAB   TEMPFAODESC, FAOPOINTER                      ; 1863
                   04   AE 00000000G    00  D0 000E1       MOVL    DIF$GL_INBUF, BUFFERPOINTER                  ; 1864
                                        05  11 000E9       BRB     7$                                           ; 1856
                   04   AE         44   AE  D0 000EB  6$:  MOVL    LINEDESC+4, BUFFERPOINTER                    ; 1866
                                   38   AE  9F 000F0  7$:  PUSHAB  OUTPUTDESC                                   ; 1871
                                        5B  DD 000F3       PUSHL   FAOPOINTER                                   ; 1872
                                        7E  D4 000F5       CLRL    -(SP)                                        ; 1871
                                        5A  DD 000F7       PUSHL   ENTSINREC                                    ; 1872
                                   10   AE  DD 000F9       PUSHL   BYTENUMBER                                   ;
                                        04  DD 000FC       PUSHL   #4                                           ; 1871
                           00000000G    00  DD 000FE       PUSHL   DIF$GL_ENTSPERLINE                           ;
```

```
                                        20 AE  DD 00104          PUSHL   BUFFERPOINTER                                          ;
                 00000000G   00                 08 FB  00107          CALLS   #8, DIF$FORMAT_HEX_OCTAL                          ;
                             60                 58 E9  0010E          BLBC    R8, 10$                                          ; 1878
                             58 00000000G       00 D0  00111          MOVL    DIF$GL_DUMPWIDTH, R8                             ; 1880
                             57           38    AE 3C  00118          MOVZWL  OUTPUTDESC, PADBYTES                             ;
                      57     58                 57 C3  0011C          SUBL3   PADBYTES, R8, PADBYTES                           ;
                             56           40    AE 3C  00120          MOVZWL  LINEDESC, ADDITIONAL                             ; 1886
                7E    00     56                 01 7A  00124          EMUL    #1, ADDITIONAL, #0, -(SP)                        ;
                56    56     8E                 04 7B  00129          EDIV    #4, (SP)+, ADDITIONAL, ADDITIONAL                ;
                                                56 D5  0012E          TSTL    ADDITIONAL                                       ;
                                                22 15  00130          BLEQ    9$                                               ;
                      50     56                 03 78  00132          ASHL    #3, ADDITIONAL, R0                               ; 1892
                      0C 00000000G   00         01 E1  00136          BBC     #1, DIF$GL_FLAGS, 8$                             ; 1888
                             50                 03 C0  0013E          ADDL2   #3, R0                                           ; 1892
                             50                 04 C6  00141          DIVL2   #4, R0                                           ;
                      56     09                 50 C3  00144          SUBL3   R0, #9, ADDITIONAL                               ;
                                                0A 11  00148          BRB     9$                                               ; 1890
                             50                 02 C0  0014A 8$:      ADDL2   #2, R0                                           ; 1896
                             50                 03 C6  0014D          DIVL2   #3, R0                                           ;
                      56     0C                 50 C3  00150          SUBL3   R0, #12, ADDITIONAL                              ;
                             57                 56 C0  00154 9$:      ADDL2   ADDITIONAL, PADBYTES                             ; 1897
                             50           38    AE 3C  00157          MOVZWL  OUTPUTDESC, R0                                   ; 1898
                             50                 56 C2  0015B          SUBL2   ADDITIONAL, R0                                   ;
          3C BE47      3C BE46                 50 28  0015E          MOVC3   R0, @OUTPUTDESC+4[ADDITIONAL], -                 ; 1900
                                                                              @OUTPUTDESC+4[PADBYTES]
                57     20           6E         00 2C  00166          MOVC5   #0, (SP), #32, PADBYTES, @OUTPUTDESC+4           ; 1901
                                          3C    BE     0016B
                                    38    AE    58 B0  0016D          MOVW    R8, OUTPUTDESC                                   ; 1902
                                    38    AE    9F 00171 10$:         PUSHAB  OUTPUTDESC                                       ; 1908
                 00000000V          EF          01 FB  00174          CALLS   #1, PUT_DESC                                     ;
                                    5A 00000000G  00 C2  00178          SUBL2   DIF$GL_ENTSPERLINE, ENTSINREC                    ; 1913
                                    6E            59 C0  00182          ADDL2   BYTESPERLINE, BYTENUMBER                         ; 1914
                                40  AE            59 A2  00185          SUBW2   BYTESPERLINE, LINEDESC                           ; 1915
                                44  AE            59 C0  00189          ADDL2   BYTESPERLINE, LINEDESC+4                         ; 1916
                                          FF0D    31 00018D             BRW     4$                                               ; 1850
                                    50          01 D0  00190 11$:      MOVL    #1, R0                                           ; 1920
                                                04 00193              RET                                                      ; 1921

; Routine Size:  404 bytes,    Routine Base:  $CODE$ + 0C2B
```

```
;  1636     1922  1 ROUTINE put_record_parallel (masrdb, revrdb) =
;  1637     1923  2 BEGIN
;  1638     1924  2
;  1639     1925  2 !++
;  1640     1926  2 !
;  1641     1927  2 ! FUNCTIONAL DESCRIPTION:
;  1642     1928  2 !
;  1643     1929  2 !       Format and put a record to the output file in parallel mode.
;  1644     1930  2 !
;  1645     1931  2 ! INPUTS:
;  1646     1932  2 !
;  1647     1933  2 !       masrdb = The address of the RDB for the master file record.
;  1648     1934  2 !
;  1649     1935  2 !       revrdb = The address of the RDB for the revision file record.
;  1650     1936  2 !
;  1651     1937  2 ! OUTPUTS:
;  1652     1938  2 !
;  1653     1939  2 !       The specified records are output in parallel mode.
;  1654     1940  2 !
;  1655     1941  2 ! ROUTINE VALUES:
;  1656     1942  2 !
;  1657     1943  2 !       Always true
;  1658     1944  2 !
;  1659     1945  2 !--
;  1660     1946  2
;  1661     1947  2 MAP
;  1662     1948  2     masrdb : REF BBLOCK,
;  1663     1949  2     revrdb : REF BBLOCK;
;  1664     1950  2
;  1665     1951  2 LOCAL
;  1666     1952  2     halfline,                               ! Amount of space for each file
;  1667     1953  2     linedesc : BBLOCK [dsc$c_s_bln],        ! Descriptor for output string
;  1668     1954  2     text_length;                            ! Amount of space left for record text
;  1669     1955  2
;  1670     1956  2 halfline = (.dif$gl_parwidth - 5)/ 2;           ! Calculate amount of space for each file
;  1671     1957  2
;  1672     1958  2 linedesc [dsc$a_pointer] = .dif$gl_outbuf;      ! Clear the output descriptor/buffer
;  1673     1959  2 CH$FILL (%ASCII' ', .dif$gl_parwidth, .linedesc [dsc$a_pointer]);
;  1674     1960  2
;  1675     1961  2 !
;  1676     1962  2 ! If masrdb is not zero, then insert text for master file.
;  1677     1963  2 ! Use exact or edited text, as appropriate.
;  1678     1964  2 !
;  1679     1965  2 IF .masrdb NEQ 0
;  1680     1966  2    THEN IF .dif$gl_ignore [ign$v_exact] AND .masrdb [rdb$v_edited]
;  1681     1967  3              THEN BEGIN
;  1682     1968  3                   linedesc [dsc$w_length] = 0;
;  1683     1969  3                   dif$gl_masfdb [fdb$l_currec] = .masrdb;
;  1684     1970  3                   get_rfa_text (dif$gl_masfdb, linedesc, .halfline);
;  1685     1971  3                   END
;  1686     1972  3              ELSE BEGIN
;  1687     1973  3                   text_length = MINU (.halfline, .masrdb [rdb$w_length]);
;  1688     1974  3                   CH$MOVE (.text_length, masrdb [rdb$t_text],
;  1689     1975  3                            .linedesc [dsc$a_pointer]);
;  1690     1976  3                   END;
;  1691     1977  2 !
;  1692     1978  2 ! Remove tabs from text and insert mid-line bar.
```

```
;  1693          1979  2 !
;  1694          1980  2 translate_tabs (.dif$gl_outbuf, .dif$gl_outbuf, .halfline, .halfline);
;  1695          1981  2 CH$WCHAR ('%ASCII ':', .dif$gl_outbuf + .halfline + 2);
;  1696          1982  2
;  1697          1983  2 !
;  1698          1984  2 ! If revrdb is not zero, then insert text for master file.
;  1699          1985  2 ! Use exact or edited text, as appropriate.
;  1700          1986  2 !
;  1701          1987  2 IF .revrdb NEQ 0
;  1702          1988  2    THEN IF .dif$gl_ignore [ign$v_exact] AND .revrdb [rdb$v_edited]
;  1703          1989  3         THEN BEGIN
;  1704          1990  3              linedesc [dsc$w_length] = .halfline + 5;
;  1705          1991  3              dif$gl_revfdb [fdb$l_currec] = .revrdb;
;  1706          1992  3              get_rfa_text (dif$gl_revfdb, linedesc, .dif$gl_parwidth);
;  1707          1993  3              END
;  1708          1994  3         ELSE BEGIN
;  1709          1995  3              text_length = MINU (.halfline, .revrdb [rdb$w_length]);
;  1710          1996  3              CH$MOVE (.text_length, revrdb [rdb$t_text],
;  1711          1997  3                       .linedesc [dsc$a_pointer] + .halfline + 5);
;  1712          1998  3              END;
;  1713          1999  2
;  1714          2000  2 !
;  1715          2001  2 ! Remove tabs from remainder of text and output the line.
;  1716          2002  2 !
;  1717          2003  2 linedesc [dsc$w_length] =
;  1718          2004  2    translate_tabs (.dif$gl_outbuf, .dif$gl_outbuf + .halfline + 5, .dif$gl_parwidth, .dif$gl_parwidth);
;  1719          2005  2 put_desc (linedesc);
;  1720          2006  2
;  1721          2007  2 RETURN true;
;  1722          2008  1 END;


                                OFFC 00000 PUT_RECORD_PARALLEL:
                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 1922
                          5B 00000000G   00  9E 00002           MOVAB   DIF$GL_MASFDB, R11              ;
                          5A 00000000G   00  9E 00009           MOVAB   DIF$GL_IGNORE, R10              ;
                          59 00000000G   00  9E 00010           MOVAB   DIF$GL_PARWIDTH, R9             ;
                          58 00000000G   00  9E 00017           MOVAB   DIF$GL_OUTBUF, R8               ;
                          5E             08  C2 0001E           SUBL2   #8, SP                          ;
                          51             69  D0 00021           MOVL    DIF$GL_PARWIDTH, R1             ; 1956
                          50       FB    A1  9E 00024           MOVAB   -5(R1), R0                      ;
                    56    50             02  C7 00028           DIVL3   #2, R0, HALFLINE                ;
                       04 AE             68  D0 0002C           MOVL    DIF$GL_OUTBUF, LINEDESC+4       ; 1958
                51  20    6E             00  2C 00030           MOVC5   #0, (SP), #32, R1, @LINEDESC+4  ; 1959
                                         04  BE 00035           ;
                          52       04    AC  D0 00037           MOVL    MASRDB, R2                      ; 1965
                                         36  13 0003B           BEQL    3$                              ;
                    1A    6A             06  E1 0003D           BBC     #6, DIF$GL_IGNORE, 1$           ; 1966
                    15 08 A2             03  E1 00041           BBC     #3, 8(R2), 1$                   ;
                                         6E  B4 00046           CLRW    LINEDESC                        ; 1968
                          6B             52  D0 00048           MOVL    R2, DIF$GL_MASFDB               ; 1969
                                         56  DD 0004B           PUSHL   HALFLINE                        ; 1970
                                   04    AE  9F 0004D           PUSHAB  LINEDESC                        ;
                                         5B  DD 00050           PUSHL   R11                             ;
```

```
                        00000000V  EF                03  FB 00052          CALLS   #3, GET_RFA_TEXT                                    ;
                                                      18  11 00059          BRB     3$                                                  ; 1966
                                   50                 56  D0 00058 1$:      MOVL    HALFLINE, R0                                        ; 1973
          50        12   A2        10                 00  ED 0005E          CMPZV   #0, #16, 18(R2), R0
                                                      04  1E 00064          BGEQU   2$
                                   50          12     A2  3C 00066          MOVZWL  18(R2), R0
                                   57                 50  D0 0006A 2$:      MOVL    R0, TEXT_LENGTH
                   04   BE   14   A2                 57  28 0006D          MOVC3   TEXT_LENGTH, 20(R2), @LINEDESC+4                    ; 1975
                                                      56  DD 00073 3$:      PUSHL   HALFLINE                                            ; 1980
                                                      56  DD 00075          PUSHL   HALFLINE
                                   50                 68  D0 00077          MOVL    DIF$GL_OUTBUF, R0
                                                      50  DD 0007A          PUSHL   R0
                                                      50  DD 0007C          PUSHL   R0
                        00000000V  EF                 04  FB 0007E          CALLS   #4, TRANSLATE_TABS
                                   50                 68  D0 00085          MOVL    DIF$GL_OUTBUF, R0                                   ; 1981
                             02 A640          7C     8F  90 00088          MOVB    #124, 2(HALFLINE)[R0]
                                   52          08     AC  D0 0008E          MOVL    REVRDB, R2                                          ; 1987
                                                      45  13 00092          BEQL    6$
                24                 6A                 06  E1 00094          BBC     #6, DIF$GL_IGNORE, 4$                               ; 1988
                1F         08      A2                 03  E1 00098          BBC     #3, 8(R2), 4$
                6E                 56                 05  A1 0009D          ADDW3   #5, HALFLINE, LINEDESC                              ; 1990
                        00000000G  00                 52  D0 000A1          MOVL    R2, DIF$GL_REVFDB                                   ; 1991
                                                      69  DD 000A8          PUSHL   DIF$GL_PARWIDTH                                     ; 1992
                                              04      AE  9F 000AA          PUSHAB  LINEDESC
                                   00000000G  00      9F  9F 000AD          PUSHAB  DIF$GL_REVFDB
                        00000000V  EF                 03  FB 000B3          CALLS   #3, GET_RFA_TEXT                                    ;
                                                      1D  11 000BA          BRB     6$                                                  ; 1988
                                   50                 56  D0 000BC 4$:      MOVL    HALFLINE, R0                                        ; 1995
          50        12   A2        10                 00  ED 000BF          CMPZV   #0, #16, 18(R2), R0
                                                      04  1E 000C5          BGEQU   5$
                                   50          12     A2  3C 000C7          MOVZWL  18(R2), R0
                                   57                 50  D0 000CB 5$:      MOVL    R0, TEXT_LENGTH
                         50        56          04     AE  C1 000CE          ADDL3   LINEDESC+4, HALFLINE, R0                            ; 1997
                   05    A0   14   A2                 57  28 000D3          MOVC3   TEXT_LENGTH, 20(R2), 5(R0)
                                   50                 69  D0 000D9 6$:      MOVL    DIF$GL_PARWIDTH, R0                                 ; 2004
                                                      50  DD 000DC          PUSHL   R0
                                                      50  DD 000DE          PUSHL   R0
                                   50                 68  D0 000E0          MOVL    DIF$GL_OUTBUF, R0
                                              05 A640 9F  9F 000E3          PUSHAB  5(HALFLINE)[R0]
                                                      50  DD 000E7          PUSHL   R0
                        00000000V  EF                 04  FB 000E9          CALLS   #4, TRANSLATE_TABS
                                   6E                 50  B0 000F0          MOVW    R0, LINEDESC
                                                      5E  DD 000F3          PUSHL   SP                                                  ; 2005
                        00000000V  EF                 01  FB 000F5          CALLS   #1, PUT_DESC
                                   50                 01  D0 000FC          MOVL    #1, R0                                              ; 2007
                                                      04  000FF            RET                                                         ; 2008

; Routine Size:  256 bytes,    Routine Base:  $CODE$ + 0DBF
```

```
;  1724          2009  1 ROUTINE translate_tabs (bufaddr, start, buflen, maxlen) =
;  1725          2010  2 BEGIN
;  1726          2011  2
;  1727          2012  2 !++
;  1728          2013  2 !
;  1729          2014  2 ! FUNCTIONAL DESCRIPTION:
;  1730          2015  2 !
;  1731          2016  2 !       Convert the tabs in the buffer, starting at start, to spaces.
;  1732          2017  2 !       Also replace line feeds, form feeds, and carriage returns with
;  1733          2018  2 !       appropriate text.
;  1734          2019  2 !
;  1735          2020  2 ! INPUTS:
;  1736          2021  2 !
;  1737          2022  2 !       bufaddr =       Address of the input buffer.
;  1738          2023  2 !
;  1739          2024  2 !       start =         Address of first char to examine.
;  1740          2025  2 !
;  1741          2026  2 !       buflen =        Length of the input text.
;  1742          2027  2 !
;  1743          2028  2 !       maxlen =        Length of buffer
;  1744          2029  2 !
;  1745          2030  2 ! OUTPUTS:
;  1746          2031  2 !
;  1747          2032  2 !       All tabs are converted to spaces.
;  1748          2033  2 !
;  1749          2034  2 ! ROUTINE VALUES:
;  1750          2035  2 !
;  1751          2036  2 !       Length of translated text
;  1752          2037  2 !
;  1753          2038  2 !--
;  1754          2039  2 LOCAL
;  1755          2040  2     charptr,                           ! Address of character to translate
;  1756          2041  2     endptr,                            ! Address of last byte to translate+1
;  1757          2042  2     maxptr,                            ! Address of last byte of buffer+1
;  1758          2043  2     fill,                              ! Number of characters to insert
;  1759          2044  2     textaddr : REF VECTOR [, BYTE];    ! Address of text to insert
;  1760          2045  2
;  1761          2046  2 charptr = .start;                                    ! Starting character
;  1762          2047  2 endptr = .bufaddr + .buflen;                         ! Last character to translate+1
;  1763          2048  2 maxptr = .bufaddr + .maxlen;                        ! End of buffer+1
;  1764          2049  2
;  1765          2050  3 WHILE (.charptr LSSA .endptr)                       ! Examine all characters
;  1766          2051  3 DO BEGIN
;  1767          2052  3    IF CH$RCHAR (.charptr) GEQU %X'20'
;  1768          2053  3         AND CH$RCHAR (.charptr) LEQU %X'7E'
;  1769          2054  3    THEN                                             ! Printing character
;  1770          2055  3         charptr = .charptr + 1
;  1771          2056  3    ELSE
;  1772          2057  4         BEGIN
;  1773          2058  4         SELECTONE CH$RCHAR (.charptr) OF
;  1774          2059  4         SET
;  1775          2060  4         [%X'09']:                                   ! Tab
;  1776          2061  5            BEGIN
;  1777          2062  5            fill = 8 - ((.charptr - .start) MOD 8);  ! Calculate number of spaces to use
;  1778          2063  5            textaddr = blanks [i];                   ! Insert blanks
;  1779          2064  4            END;
;  1780          2065  4
```

```
;  1781     2066  4            [%X'0A']:                                  ! Line feed
;  1782     2067  5               BEGIN
;  1783     2068  5               textaddr = lf;                          ! Insert <LF>
;  1784     2069  5               fill = .textaddr [0];                   ! Length of string
;  1785     2070  5               textaddr = textaddr [1];                ! Text address
;  1786     2071  4               END;
;  1787     2072  4
;  1788     2073  4            [%X'0B']:                                  ! Vertical tab
;  1789     2074  5               BEGIN
;  1790     2075  5               textaddr = vt;                          ! Insert <CR>
;  1791     2076  5               fil  = .textaddr [0];                   ! Length of string
;  1792     2077  5               textaddr = textaddr [1];                ! Text address
;  1793     2078  4               END;
;  1794     2079  4
;  1795     2080  4            [%X'0C']:                                  ! Form feed
;  1796     2081  5               BEGIN
;  1797     2082  5               textaddr = ff;                          ! Insert <FF>
;  1798     2083  5               fill = .textaddr [0];                   ! Length of string
;  1799     2084  5               textaddr = textaddr [1];                ! Text address
;  1800     2085  4               END;
;  1801     2086  4
;  1802     2087  4            [%X'0D']:                                  ! Carriage return
;  1803     2088  5               BEGIN
;  1804     2089  5               textaddr = cr;                          ! Insert <CR>
;  1805     2090  5               fill = .textaddr [0];                   ! Length of string
;  1806     2091  5               textaddr = textaddr [1];                ! Text address
;  1807     2092  4               END;
;  1808     2093  4
;  1809     2094  4            [OTHERWISE]:                               ! All other characters
;  1810     2095  5               BEGIN
;  1811     2096  5               textaddr = period;                      ! Insert period
;  1812     2097  5               fill = .textaddr [0];                   ! Length of string
;  1813     2098  5               textaddr = textaddr [1];                ! Text address
;  1814     2099  4               END;
;  1815     2100  4
;  1816     2101  4            TES;
;  1817     2102  4
;  1818     2103  4            IF .fill GTR .maxptr - .charptr             ! Shorten it if it goes past end of buffer
;  1819     2104  4               THEN fill = .maxptr - .charptr;
;  1820     2105  4            endptr = CH$MOVE (                         ! Shift the unexamined text over
;  1821     2106  4                   MIN (                               ! and update endptr
;  1822     2107  4                           .maxptr - .charptr - .fill, ! Target space available
;  1823     2108  4                           .endptr - .charptr - 1),    ! Number of source characters remaining
;  1824     2109  4                   .charptr+1,
;  1825     2110  4                   .charptr + .fill);
;  1826     2111  4            charptr = CH$MOVE (.fill, .textaddr, .charptr); ! Insert the text and update charptr
;  1827     2112  3            END;
;  1828     2113  2        END;
;  1829     2114  2
;  1830     2115  2 RETURN .charptr-.bufaddr;
;  1831     2116  1 END;


                                OFFC 00000 TRANSLATE_TABS:
```

```
                                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                    ; 2009
                                5B 00000000'   EF   9E 00002                    MOVAB   BLANKS+1, R11
                                58       08    AC   D0 00009                    MOVL    START, CHARPTR                                          ; 2046
                          5A    04       0C    AC   C1 0000D                    ADDL3   BUFLEN, BUFADDR, ENDPTR                                 ; 2047
                          57    04       10    AC   C1 00013                    ADDL3   MAXLEN, BUFADDR, MAXPTR                                 ; 2048
                                5A             58   D1 00019 1$:               CMPL    CHARPTR, ENDPTR                                         ; 2050
                                               03   1F 0001C                    BLSSU   2$
                                           008F  31   0001E                    BRW     13$
                                20             68   91 00021 2$:               CMPB    (CHARPTR), #32                                          ; 2052
                                               0A   1F 00024                    BLSSU   3$
                                7E   8F        68   91 00026                    CMPB    (CHARPTR), #126                                         ; 2053
                                               04   1A 0002A                    BGTRU   3$
                                               58   D6 0002C                    INCL    CHARPTR                                                 ; 2055
                                               E9   11 0002E                    BRB     1$
                                09             68   91 00030 3$:               CMPB    (CHARPTR), #9                                           ; 2060
                                               18   12 00033                    BNEQ    4$
                          50    08       AC    58   C3 00035                    SUBL3   CHARPTR, START, R0                                      ; 2062
                    7E    00             50    01   7A 0003A                    EMUL    #1, R0, #0, -(SP)
                    50    50             8E    08   7B 0003F                    EDIV    #8, (SP)+, R0, R0
                                56       08    A0   9E 00044                    MOVAB   8(R0), FILL
                                59             6B   9E 00048                    MOVAB   BLANKS+1, TEXTADDR                                      ; 2063
                                               33   11 0004B                    BRB     10$                                                     ; 2058
                                0A             68   91 0004D 4$:               CMPB    (CHARPTR), #10                                          ; 2066
                                               06   12 00050                    BNEQ    5$
                                59       EF    AB   9E 00052                    MOVAB   LF, TEXTADDR                                            ; 2068
                                               25   11 00056                    BRB     9$                                                      ; 2069
                                0B             68   91 00058 5$:               CMPB    (CHARPTR), #11                                          ; 2073
                                               06   12 0005B                    BNEQ    6$
                                59       F7    AB   9E 0005D                    MOVAB   VT, TEXTADDR                                            ; 2075
                                               1A   11 00061                    BRB     9$                                                      ; 2076
                                0C             68   91 00063 6$:               CMPB    (CHARPTR), #12                                          ; 2080
                                               06   12 00066                    BNEQ    7$
                                59       E7    AB   9E 00068                    MOVAB   FF, TEXTADDR                                            ; 2082
                                               0F   11 0006C                    BRB     9$                                                      ; 2083
                                0D             68   91 0006E 7$:               CMPB    (CHARPTR), #13                                          ; 2087
                                               06   12 00071                    BNEQ    8$
                                59       DF    AB   9E 00073                    MOVAB   CR, TEXTADDR                                            ; 2089
                                               04   11 00077                    BRB     9$                                                      ; 2090
                                59       0B    AB   9E 00079 8$:               MOVAB   PERIOD, TEXTADDR                                        ; 2096
                                56             89   9A 0007D 9$:               MOVZBL  (TEXTADDR)+, FILL                                       ; 2097
                          50    57             58   C3 00080 10$:              SUBL3   CHARPTR, MAXPTR, R0                                     ; 2103
                                50             56   D1 00084                    CMPL    FILL, R0
                                               03   15 00087                    BLEQ    11$
                                56             50   D0 00089                    MOVL    R0, FILL                                                ; 2104
                                50             56   C2 0008C 11$:              SUBL2   FILL, R0                                                ; 2107
                          53    5A             58   C3 0008F                    SUBL3   CHARPTR, ENDPTR, R3                                     ; 2108
                                               53   D7 00093                    DECL    R3
                                53             50   D1 00095                    CMPL    R0, R3
                                               03   15 00098                    BLEQ    12$
                                50             53   D0 0009A                    MOVL    R3, R0
                    6648  01    A8             50   28 0009D 12$:              MOVC3   R0, 1(CHARPTR), (FILL)[CHARPTR]                         ; 2110
                                5A             53   D0 000A3                    MOVL    R3, ENDPTR
                          68    69             56   28 000A6                    MOVC3   FILL, (TEXTADDR), (CHARPTR)                             ; 2111
                                58             53   D0 000AA                    MOVL    R3, CHARPTR
                                           FF69  31   000AD                    BRW     1$                                                      ; 2050
                          53    58       04    AC   C3 000B0 13$:              SUBL3   BUFADDR, CHARPTR, R3                                    ; 2115
                                50             53   D0 000B5                    MOVL    R3, R0
```

```
                                04 000B8        RET                                     ; 2116

; Routine Size:  185 bytes,    Routine Base:  $CODE$ + 0EBF
```

```
;  1833          2117  1 ROUTINE put_hex_octal_header (number, length, cbarflag) =
;  1834          2118  2 BEGIN
;  1835          2119  2
;  1836          2120  2 !++
;  1837          2121  2 !
;  1838          2122  2 ! FUNCTIONAL DESCRIPTION:
;  1839          2123  2 !
;  1840          2124  2 !       Put a hex or octal record header to the output file.
;  1841          2125  2 !
;  1842          2126  2 ! INPUTS:
;  1843          2127  2 !
;  1844          2128  2 !       number =   The number of the record.
;  1845          2129  2 !
;  1846          2130  2 !       length =   The length of the record in bytes.
;  1847          2131  2 !
;  1848          2132  2 !       cbarflag =  A flag that is 0 if not changebar format
;  1849          2133  2 !                                  1 if changebar format and bar should be output
;  1850          2134  2 !                                  2 if changebar format and bar should not be ouput
;  1851          2135  2 !
;  1852          2136  2 !
;  1853          2137  2 ! OUTPUTS:
;  1854          2138  2 !
;  1855          2139  2 !       None
;  1856          2140  2 !
;  1857          2141  2 ! ROUTINE VALUES:
;  1858          2142  2 !
;  1859          2143  2 !       Always true
;  1860          2144  2 !
;  1861          2145  2 !--
;  1862          2146  2 LOCAL
;  1863          2147  2     changedesc : BBLOCK [dsc$c_s_bln],                  ! Desc for change text
;  1864          2148  2     faodesc : BBLOCK [dsc$c_s_bln],                     ! Desc for fao control string
;  1865          2149  2     linedesc : BBLOCK [dsc$c_s_bln];                    ! Desc for output string
;  1866          2150  2
;  1867          2151  2 linedesc [dsc$w_length] = .dif$gl_dumpwidth;          ! Init output desc
;  1868          2152  2 linedesc [dsc$a_pointer] = .dif$gl_outbuf;
;  1869          2153  2
;  1870          2154  2 IF .dif$gl_flags [dif$v_hex]                                ! Init fao desc
;  1871          2155  3    THEN BEGIN
;  1872          2156  3         faodesc [dsc$w_length] = .hexheader [0];
;  1873          2157  3         faodesc [dsc$a_pointer] = hexheader [1];
;  1874          2158  3         END
;  1875          2159  3    ELSE BEGIN
;  1876          2160  3         faodesc [dsc$w_length] = .octheader [0];
;  1877          2161  3         faodesc [dsc$a_pointer] = octheader [1];
;  1878          2162  3         END;
;  1879          2163  2
;  1880          2164  2 changedesc [dsc$a_pointer] = change [1];                  ! Init change desc
;  1881          2165  2 IF .cbarflag
;  1882          2166  2    THEN changedesc [dsc$w_length] = .change [0]
;  1883          2167  2    ELSE changedesc [dsc$w_length] = 0;
;  1884          2168  2
;  1885          2169  2 !
;  1886          2170  2 ! Generate output string and write it to the file.
;  1887          2171  2 !
;  1888          2172  2 SYS$FAO (faodesc, linedesc, linedesc, .number, .number, .length, .length, changedesc);
;  1889          2173  2 put_desc (linedesc);
```

```
;  1890          2174  2
;  1891          2175  2 RETURN true;
;  1892          2176  1 END;


                                        0004 00000 PUT_HEX_OCTAL_HEADER:
                                                   .WORD   Save R2                                          ; 2117
                          52 00000000'  EF  9E 00002       MOVAB   HEXHEADER, R2                          ;
                          5E            18  C2 00009       SUBL2   #24, SP                                ;
                          6E 00000000G  00  B0 0000C       MOVW    DIF$GL_DUMPWIDTH, LINEDESC             ; 2151
                       04 AE 00000000G  00  D0 00013       MOVL    DIF$GL_OUTBUF, LINEDESC+4              ; 2152
                0B 00000000G  00        01  E1 0001B       BBC     #1, DIF$GL_FLAGS, 1$                   ; 2154
                          08 AE         62  9B 00023       MOVZBW  HEXHEADER, FAODESC                     ; 2156
                          0C AE    01   A2  9E 00027       MOVAB   HEXHEADER+1, FAODESC+4                 ; 2157
                                        0A  11 0002C       BRB     2$                                     ; 2154
                          08 AE    6C   A2  9B 0002E 1$:    MOVZBW  OCTHEADER, FAODESC                     ; 2160
                          0C AE    6D   A2  9E 00033       MOVAB   OCTHEADER+1, FAODESC+4                 ; 2161
                          14 AE  FF79   C2  9E 00038 2$:    MOVAB   CHANGE+1, CHANGEDESC+4                 ; 2164
                          08     0C     AC  E9 0003E       BLBC    CBARFLAG, 3$                           ; 2165
                          10 AE  FF78   C2  9B 00042       MOVZBW  CHANGE, CHANGEDESC                     ; 2166
                                        03  11 00048       BRB     4$                                     ;
                                  10 AE B4  0004A 3$:      CLRW    CHANGEDESC                             ; 2167
                                  10 AE 9F  0004D 4$:      PUSHAB  CHANGEDESC                             ; 2172
                                  08 AC DD  00050          PUSHL   LENGTH                                 ;
                             7E   04 AC 7D  00053          MOVQ    NUMBER, -(SP)                          ;
                                  04 AC DD  00057          PUSHL   NUMBER                                 ;
                                  14 AE 9F  0005A          PUSHAB  LINEDESC                               ;
                                  18 AE 9F  0005D          PUSHAB  LINEDESC                               ;
                                  24 AE 9F  00060          PUSHAB  FAODESC                                ;
                   00000000G  00        08  FB 00063       CALLS   #8, SYS$FAO                            ;
                                        5E  DD 0006A       PUSHL   SP                                     ; 2173
                   00000000V  EF        01  FB 0006C       CALLS   #1, PUT_DESC                           ;
                              50        01  D0 00073       MOVL    #1, R0                                 ; 2175
                                        04  00076       RET                                            ; 2176

; Routine Size:  119 bytes,    Routine Base:  $CODE$ + 0F78
```

```
;  1894          2177  1 ROUTINE put_idline (fdb) =
;  1895          2178  2 BEGIN
;  1896          2179  2
;  1897          2180  2 !++
;  1898          2181  2 !
;  1899          2182  2 ! FUNCTIONAL DESCRIPTION:
;  1900          2183  2 !
;  1901          2184  2 !       Put a record containing file id line to the output file.
;  1902          2185  2 !
;  1903          2186  2 ! INPUTS:
;  1904          2187  2 !
;  1905          2188  2 !       fdb =   The address of the FDB of the file to be described.
;  1906          2189  2 !
;  1907          2190  2 ! OUTPUTS:
;  1908          2191  2 !
;  1909          2192  2 !       None
;  1910          2193  2 !
;  1911          2194  2 ! ROUTINE VALUES:
;  1912          2195  2 !
;  1913          2196  2 !       Always true
;  1914          2197  2 !
;  1915          2198  2 !--
;  1916          2199  2
;  1917          2200  2 MAP
;  1918          2201  2     fdb : REF BBLOCK;
;  1919          2202  2
;  1920          2203  2 LOCAL
;  1921          2204  2     linedesc : BBLOCK [dsc$c_s_bln];            ! Descriptor for the output string
;  1922          2205  2
;  1923          2206  2 BIND
;  1924          2207  2     filedesc = fdb [fdb$l_fildesc] : REF BBLOCK [dsc$c_s_bln];  ! resultant file name string descriptor
;  1925          2208  2
;  1926          2209  2 linedesc [dsc$a_pointer] = .dif$gl_outbuf;                   ! Init output desc pointer to buffer
;  1927          2210  2 linedesc [dsc$w_length] = MINU (                            ! Set output desc size
;  1928          2211  2     .file [0] + .filedesc [dsc$w_length],
;  1929          2212  2     .dif$gl_width);
;  1930          2213  2
;  1931          2214  2 CH$COPY (
;  1932          2215  2     .file [0], file [1],                                    ! 'FILE' string
;  1933          2216  2     .filedesc [dsc$w_length], .filedesc [dsc$a_pointer],    ! resultant file name
;  1934          2217  2     0,                                                      ! fill never used
;  1935          2218  2     .linedesc [dsc$w_length], .linedesc [dsc$a_pointer]);   ! line buffer
;  1936          2219  2
;  1937          2220  2 put_desc (linedesc);                                       ! Output id line
;  1938          2221  2
;  1939          2222  2 RETURN true;
;  1940          2223  1 END;


                                OFFC 00000 PUT_IDLINE:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 2177
                5B 00000000'    EF  9E 00002       MOVAB   FILE, R11                           ;
                5E              08  C2 00009       SUBL2   #8, SP                              ;
     50         04   AC         38  C1 0000C       ADDL3   #56, FDB, R0                        ; 2207
```

```
                        04   AE 00000000G   00   D0 00011          MOVL    DIF$GL_OUTBUF, LINEDESC+4                    ; 2209
                             57             60   D0 00019          MOVL    (R0), R7                                     ; 2211
                             50             6B   9A 0001C          MOVZBL  FILE, R0                                     ;
                             51             67   3C 0001F          MOVZWL  (R7), R1                                     ;
                             50             51   C0 00022          ADDL2   R1, R0                                       ;
                 00000000G   00             50   D1 00025          CMPL    R0, DIF$GL_WIDTH                             ; 2212
                                            07   1B 0002C          BLEQU   1$                                           ;
                             50 00000000G   00   D0 0002E          MOVL    DIF$GL_WIDTH, R0                             ;
                             6E             50   B0 00035 1$:      MOVW    R0, LINEDESC                                 ; 2210
                             5A             6B   9A 00038          MOVZBL  FILE, R10                                    ; 2215
                             59             67   3C 0003B          MOVZWL  (R7), R9                                     ; 2216
                             58             6E   3C 0003E          MOVZWL  LINEDESC, R8                                 ; 2218
                             56       04    AE   D0 00041          MOVL    LINEDESC+4, R6                               ;
        58         00        01       AB    5A   2C 00045          MOVC5   R10, FILE+1, #0, R8, (R6)                    ;
                                             66        0004B                                                           ;
                                             0D   18 0004C          BGEQ    2$                                           ;
                             56              5A   C0 0004E          ADDL2   R10, R6                                      ;
                             58              5A   C2 00051          SUBL2   R10, R8                                      ;
        58         00        04       B7    59   2C 00054          MOVC5   R9, @4(R7), #0, R8, (R6)                     ;
                                             66        0005A                                                           ;
                                             5E   DD 0005B 2$:      PUSHL   SP                                           ; 2220
                 00000000V   EF             01   FB 0005D          CALLS   #1, PUT_DESC                                 ;
                             50              01   D0 00064          MOVL    #1, R0                                       ; 2222
                                             04   00067            RET                                                  ; 2223

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 0FEF
```

```
;  1942          2224  1 ROUTINE put_parallel_idline =
;  1943          2225  2 BEGIN
;  1944          2226  2
;  1945          2227  2 !++
;  1946          2228  2 !
;  1947          2229  2 ! FUNCTIONAL DESCRIPTION:
;  1948          2230  2 !
;  1949          2231  2 !       Put a record containing both file ids to the output file,
;  1950          2232  2 !       in parallel format.
;  1951          2233  2 !
;  1952          2234  2 ! INPUTS:
;  1953          2235  2 !
;  1954          2236  2 !       None
;  1955          2237  2 !
;  1956          2238  2 ! OUTPUTS:
;  1957          2239  2 !
;  1958          2240  2 !       None
;  1959          2241  2 !
;  1960          2242  2 ! ROUTINE VALUES:
;  1961          2243  2 !
;  1962          2244  2 !       Always true
;  1963          2245  2 !
;  1964          2246  2 !--
;  1965          2247  2
;  1966          2248  2 LOCAL
;  1967          2249  2     charptr,                                  ! Pointer into the output descriptor
;  1968          2250  2     linedesc : BBLOCK [dsc$c_s_bln],          ! Descriptor for the output string
;  1969          2251  2     namesize;                                 ! Space left for each file name
;  1970          2252  2
;  1971          2253  2 dif$gl_parwidth = 2 * ((.dif$gl_width - 1)/ 2) + 1;          ! Calculate width of parallel listing
;  1972          2254  2 linedesc [dsc$w_length] = .dif$gl_parwidth;                  ! Init output descriptor
;  1973          2255  2 linedesc [dsc$a_pointer] = .dif$gl_outbuf;
;  1974          2256  2 namesize = (.dif$gl_parwidth - 5)/ 2;                       ! Calculate space left for each file name
;  1975          2257  2
;  1976          2258  2 !
;  1977          2259  2 ! Output a line of dashes.
;  1978          2260  2 !
;  1979          2261  2 CH$FILL (%ASCII '-', .linedesc [dsc$w_length], .linedesc [dsc$a_pointer]);
;  1980          2262  2 put_desc (linedesc);
;  1981          2263  2
;  1982          2264  2 !
;  1983          2265  2 ! Build line with both file names.
;  1984          2266  2 !
;  1985          2267  2 charptr = CH$COPY (
;  1986          2268  2     .file [0],                                ! Insert 'FILE'
;  1987          2269  2     file [1],
;  1988          2270  2     .dif$gl_masdesc [dsc$w_length],           ! Insert master file name
;  1989          2271  2     .dif$gl_masdesc [dsc$a_pointer],
;  1990          2272  2     %ASCII ' ',                               ! blank fill
;  1991          2273  2     .namesize,                                ! buffer length
;  1992          2274  2     .linedesc [dsc$a_pointer]);               ! buffer address
;  1993          2275  2 CH$WCHAR_A (%ASCII ' ', charptr);
;  1994          2276  2 CH$WCHAR_A (%ASCII ' ', charptr);
;  1995          2277  2 CH$WCHAR_A (%ASCII '|', charptr);             ! insert mid-line bar
;  1996          2278  2 CH$WCHAR_A (%ASCII ' ', charptr);
;  1997          2279  2 CH$WCHAR_A (%ASCII ' ', charptr);
;  1998          2280  2 CH$COPY (
```

```
;1999          2281  2      .file [0],                          ! Insert 'FILE'
;2000          2282  2      file [1],
;2001          2283  2      .dif$gl_revdesc [dsc$w_length],     ! Insert revision file name
;2002          2284  2      .dif$gl_revdesc [dsc$a_pointer],
;2003          2285  2      %ASCII ' ',                         ! blank fill
;2004          2286  2      .namesize,                          ! buffer length
;2005          2287  2      .charptr);                          ! buffer address
;2006          2288  2
;2007          2289  2 put_desc (linedesc);                    ! Output the line
;2008          2290  2
;2009          2291  2 RETURN true;
;2010          2292  1 END;


                                                  0FFC 00000 PUT_PARALLEL_IDLINE:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11                ; 2224
                                  5E                     08  C2 00002          SUBL2   #8, SP                              ;
                        50 00000000G                     01  C3 00005          SUBL3   #1, DIF$GL_WIDTH, R0                ; 2253
                                  50                     02  C6 0000D          DIVL2   #2, R0                              ;
                00000000G  00     50                     01  78 00010          ASHL    #1, R0, DIF$GL_PARWIDTH             ;
                                        00000000G        00  D6 00018          INCL    DIF$GL_PARWIDTH                     ;
                                  50 00000C0G            00  D0 0001E          MOVL    DIF$GL_PARWIDTH, R0                 ; 2254
                                  6E                     50  B0 00025          MOVW    R0, LINEDESC                        ;
                                        04  AE 00000000G 00  D0 00028          MOVL    DIF$GL_OUTBUF, LINEDESC+4           ; 2255
                                  50                     05  C2 00030          SUBL2   #5, R0                              ; 2256
                        56        50                     02  C7 00033          DIVL3   #2, R0, NAMESIZE                    ;
        6E              2D        6E                     00  2C 00037          MOVC5   #0, (SP), #45, LINEDESC, @LINEDESC+4 ; 2261
                                        04               BE      0003C          ;                                           ;
                                  5E                     DD  0003E          PUSHL   SP                                  ; 2262
                     00000000V    EF                     01  FB 00040          CALLS   #1, PUT_DESC                        ;
                                  5B 00000000'           EF  9A 00047          MOVZBL  FILE, R11                           ; 2268
                                  5A 00000000G           00  3C 0004E          MOVZWL  DIF$GL_MASDESC, R10                 ; 2270
                                  59 00000000G           00  D0 00055          MOVL    DIF$GL_MASDESC+4, R9                ; 2271
                                  58                     56  D0 0005C          MOVL    NAMESIZE, R8                        ; 2273
                                  57        04           AE  D0 0005F          MOVL    LINEDESC+4, R7                      ; 2274
        58        20 00000000'    EF                     5B  2C 00063          MOVC5   R11, FILE+1, #32, R8, (R7)          ;
                                                         67      0006C          ;                                           ;
                                                         0C  18 0006D          BGEQ    1$                                  ;
                                  57                     5B  C0 0006F          ADDL2   R11, R7                             ;
                                  58                     5B  C2 00072          SUBL2   R11, R8                             ;
        58        20              69                     5A  2C 00075          MOVC5   R10, (R9), #32, R8, (R7)            ;
                                                         67      0007A          ;                                           ;
                                  83 207C2020            8F  D0 0007B 1$:      MOVL    #545005600, (CHARPTR)+              ; 2275
                                  83                     20  90 00082          MOVB    #32, (CHARPTR)+                     ; 2279
                                  5A 00000000'           EF  9A 00085          MOVZBL  FILE, R10                           ; 2281
                                  59 00000000G           00  3C 0008C          MOVZWL  DIF$GL_REVDESC, R9                  ; 2283
                                  58 00000000G           00  D0 00093          MOVL    DIF$GL_REVDESC+4, R8                ; 2284
                                  57                     53  D0 0009A          MOVL    CHARPTR, R7                         ; 2287
        56        20 00000000'    EF                     5A  2C 0009D          MOVC5   R10, FILE+1, #32, R6, (R7)          ;
                                                         67      000A6          ;                                           ;
                                                         0C  18 000A7          BGEQ    2$                                  ;
                                  57                     5A  C0 000A9          ADDL2   R10, R7                             ;
                                  56                     5A  C2 000AC          SUBL2   R10, R6                             ;
        56        20              68                     59  2C 000AF          MOVC5   R9, (R8), #32, R6, (R7)             ;
```

```
                                        67      000B4
                                        5E  DD 000B5 2$:     PUSHL   SP                                              ;
                                        01  FB 000B7         CALLS   #1, PUT_DESC                                    ; 2289
                            00000000V   EF                  ;
                                        50  01  D0 000BE     MOVL    #1, R0                                          ; 2291
                                            04  000C1       RET                                                     ; 2292

; Routine Size:  194 bytes,    Routine Base:  $CODE$ + 1057
```

```
;  2012          2293  1 ROUTINE put_blank =
;  2013          2294  2 BEGIN
;  2014          2295  2
;  2015          2296  2 !++
;  2016          2297  2 !
;  2017          2298  2 ! FUNCTIONAL DESCRIPTION:
;  2018          2299  2 !
;  2019          2300  2 !       Put a blank line to the output file.
;  2020          2301  2 !
;  2021          2302  2 ! INPUTS:
;  2022          2303  2 !
;  2023          2304  2 !       None
;  2024          2305  2 !
;  2025          2306  2 ! OUTPUTS:
;  2026          2307  2 !
;  2027          2308  2 !       None
;  2028          2309  2 !
;  2029          2310  2 ! ROUTINE VALUES:
;  2030          2311  2 !
;  2031          2312  2 !       Always true
;  2032          2313  2 !
;  2033          2314  2 !--
;  2034          2315  2
;  2035          2316  2 LOCAL
;  2036          2317  2     linedesc : BBLOCK [dsc$c_s_bln],
;  2037          2318  2     tempbuf;
;  2038          2319  2
;  2039          2320  2 linedesc [dsc$w_length] = 0;
;  2040          2321  2 linedesc [dsc$a_pointer] = tempbuf;
;  2041          2322  2
;  2042          2323  2 put_desc (linedesc);
;  2043          2324  2
;  2044          2325  2 RETURN true;
;  2045          2326  1 END;


                                0000 00000 PUT_BLANK:
                                           .WORD   Save nothing                    ; 2293
                           5E   0C   C2 00002      SUBL2   #12, SP                 ;
                        04 AE   B4   CF 00005      CLRW    LINEDESC                ; 2320
                     08 AE      6E   9E 00008      MOVAB   TEMPBUF, LINEDESC+4     ; 2321
                        04 AE   9F   FB 0000C      PUSHAB  LINEDESC                ; 2323
                00000000V EF    01   FB 0000F      CALLS   #1, PUT_DESC            ;
                           50   01   D0 00016      MOVL    #1, R0                  ; 2325
                                04   04 00019      RET                             ; 2326

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 1119
```

```
;  2047          2327  1 ROUTINE put_desc (linedesc) =
;  2048          2328  1 BEGIN
;  2049          2329  1
;  2050          2330  1 !++
;  2051          2331  1 ! FUNCTIONAL DESCRIPTION:
;  2052          2332  1 !
;  2053          2333  1 !       Put a descriptor to the output file.
;  2054          2334  1 !
;  2055          2335  1 ! INPUTS:
;  2056          2336  1 !
;  2057          2337  1 !       linedesc =      The address of the string descriptor for the text
;  2058          2338  1 !                       that is to be output.
;  2059          2339  1 !
;  2060          2340  1 ! OUTPUTS:
;  2061          2341  1 !
;  2062          2342  1 !       None
;  2063          2343  1 !
;  2064          2344  1 ! ROUTINE VALUES:
;  2065          2345  1 !
;  2066          2346  1 !       Always true
;  2067          2347  1 !
;  2068          2348  1 !--
;  2069          2349  1
;  2070          2350  1
;  2071          2351  1 MAP
;  2072          2352  1     linedesc : REF BBLOCK;
;  2073          2353  1
;  2074          2354  1 LOCAL
;  2075          2355  1     status;
;  2076          2356  1
;  2077          2357  1 dif$gl_outrab [rab$w_rsz] = .linedesc [dsc$w_length];
;  2078          2358  1 dif$gl_outrab [rab$l_rbf] = .linedesc [dsc$a_pointer];
;  2079          2359  1
;  2080          2360  2 IF NOT (status = $PUT (RAB = dif$gl_outrab))
;  2081          2361  1     THEN SIGNAL (dif$_writeerr, 1, dif$gl_outdesc, .status,
;  2082          2362  1                  .dif$gl_outrab [rab$l_stv]);
;  2083          2363  1
;  2084          2364  1 RETURN true;
;  2085          2365  0 END;


                                                        .EXTRN  SYS$PUT

                                        0004 00000 PUT_DESC:
                                                        .WORD   Save R2                                 ;  2327
                    52 00000000G   00    9E 00002           MOVAB   DIF$GL_OUTRAB+34, R2                ;
                    50       04    AC    D0 00009           MOVL    LINEDESC, R0                        ;  2357
                    62             60    B0 0000D           MOVW    (R0), DIF$GL_OUTRAB+34              ;
                06  A2       04    A0    D0 00010           MOVL    4(R0), DIF$GL_OUTRAB+40             ;  2358
                             DE    A2    9F 00015           PUSHAB  DIF$GL_OUTRAB                       ;  2360
            00000000G  00          01    FB 00018           CALLS   #1, SYS$PUT                         ;
                    1A             50    E8 0001F           BLBS    STATUS, 1$                          ;
                             EA    A2    DD 00022           PUSHL   DIF$GL_OUTRAB+12                    ;  2362
                                   50    DD 00025           PUSHL   STATUS                              ;  2361
                       00000000G   00    9F 00027           PUSHAB  DIF$GL_OUTDESC                      ;
                                   01    DD 0002D           PUSHL   #1                                  ;
```

```
                    00000000G  BF   DD 0002F          PUSHL   #DIF$_WRITEERR                        ;
          00000000G  00        05   FB 00035          CALLS   #5, LIB$SIGNAL                       ;
                     50        01   D0 0003C 1$:      MOVL    #1, R0                               ; 2364
                               04 0003F               RET                                          ; 2365

; Routine Size:  64 bytes,    Routine Base:  $CODE$ + 1133
```

```
;  2087          2366  1 ROUTINE insert_linenum (rdb, linedesc, condense) =
;  2088          2367  2 BEGIN
;  2089          2368  2
;  2090          2369  2 !++
;  2091          2370  2 !
;  2092          2371  2 ! FUNCTIONAL DESCRIPTION:
;  2093          2372  2 !
;  2094          2373  2 !       Insert the line number of the specified record in the output buffer.
;  2095          2374  2 !
;  2096          2375  2 ! INPUTS:
;  2097          2376  2 !
;  2098          2377  2 !       rdb =       The address of the RDB of the record whose
;  2099          2378  2 !                   line number is to be inserted.
;  2100          2379  2 !
;  2101          2380  2 !       linedesc =  The address of a string descriptor for the string that
;  2102          2381  2 !                   the number is to be inserted at the end of.
;  2103          2382  2 !
;  2104          2383  2 !       condense =  A flag that is true if the inserted number should not
;  2105          2384  2 !                   be padded with more than one blank one each side.
;  2106          2385  2 !                   The default is to pad the number out to DIF$C_LINENUM
;  2107          2386  2 !                   spaces.
;  2108          2387  2 !
;  2109          2388  2 ! OUTPUTS:
;  2110          2389  2 !
;  2111          2390  2 !       The line number is inserted in the output buffer, at the specified
;  2112          2391  2 !       position.
;  2113          2392  2 !
;  2114          2393  2 ! ROUTINE VALUES:
;  2115          2394  2 !
;  2116          2395  2 !       Always true
;  2117          2396  2 !
;  2118          2397  2 !--
;  2119          2398  2
;  2120          2399  2 MAP
;  2121          2400  2     rdb : REF BBLOCK,
;  2122          2401  2     linedesc : REF BBLOCK;
;  2123          2402  2
;  2124          2403  2 LOCAL
;  2125          2404  2     numdesc : BBLOCK [dsc$c_s_bln],                    ! Descriptor for string to contain numeric t
;  2126          2405  2     numbuf : BBLOCK [dif$c_linenum];                   ! Buffer to contain numeric text
;  2127          2406  2
;  2128          2407  2 numdesc [dsc$b_class] = dsc$k_class_s;                ! Init RTL output aescriptor
;  2129          2408  2 numdesc [dsc$w_length] = dif$c_linenum - 1;
;  2130          2409  2 numdesc [dsc$a_pointer] = numbuf;
;  2131          2410  2
;  2132          2411  2 OTS$CVT_L_TI (rdb [rdb$l_number], numdesc);             ! Insert line number in buffer
;  2133          2412  2
;  2134          2413  2 IF .condense                                           ! If inserting in parallel line
;  2135          2414  3    THEN BEGIN                                          ! Then put one blank on each side of number
;  2136          2415  3
;  2137          2416  4         WHILE (CH$RCHAR (.numdesc [dsc$a_pointer]) EQL %X '20')
;  2138          2417  4         DO BEGIN
;  2139          2418  4            numdesc [dsc$a_pointer] = .numdesc [dsc$a_pointer] + 1;
;  2140          2419  4            numdesc [dsc$w_length] = .numdesc [dsc$w_length] - 1;
;  2141          2420  3            END;
;  2142          2421  3
;  2143          2422  3         CH$WCHAR (%ASCII ' ', .linedesc [dsc$a_pointer]
```

```
;  2144          2423  3                                + .linedesc [dsc$w_length]);
;  2145          2424  3          linedesc [dsc$w_length] = .linedesc [dsc$w_length] + 1;
;  2146          2425  2          END;
;  2147          2426  2
;  2148          2427  2      CH$MOVE (.numdesc [dsc$w_length], .numdesc [dsc$a_pointer],
;  2149          2428  2               .linedesc [dsc$w_length] + .linedesc [dsc$a_pointer]);
;  2150          2429  2      CH$WCHAR (%C' ', .linedesc [dsc$a_pointer] + .linedesc [dsc$w_length]
;  2151          2430  2                     + .numdesc [dsc$w_length]);
;  2152          2431  2      linedesc [dsc$w_length] = .linedesc [dsc$w_length]
;  2153          2432  2                              + .numdesc [dsc$w_length] + 1;
;  2154          2433  2
;  2155          2434  2      RETURN true;
;  2156          2435  1      END;


                                00FC 00000 INSERT_LINENUM:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 2366
                          5E            10  C2 00002          SUBL2   #16, SP                 ;
                    0B    AE            01  90 00005          MOVB    #1, NUMDESC+3           ; 2407
                    08    AE            05  B0 00009          MOVW    #5, NUMDESC             ; 2408
                    0C    AE            6E  9E 0000D          MOVAB   NUMBUF, NUMDESC+4       ; 2409
                                08      AE  9F 00011          PUSHAB  NUMDESC                 ; 2411
               7E         04    AC        04  C1 00014          ADDL3   #4, R0B, -(SP)          ;
                  00000000G   00          02  FB 00019          CALLS   #2, OTS$CVT_L_TI        ;
                              1E    0C    AC  E9 00020          BLBC    CONDENSE, 3$            ; 2413
                              20    0C    BE  91 00024 1$:      CMPB    @NUMDESC+4, #32         ; 2416
                                          08  12 00028          BNEQ    2$                      ;
                                    0C    AE  D6 0002A          INCL    NUMDESC+4               ; 2418
                                    08    AE  B7 0002D          DECW    NUMDESC                 ; 2419
                                          F2  11 00030          BRB     1$                      ; 2416
                                50        08  AC  D0 00032 2$:      MOVL    LINEDESC, R0            ; 2422
                                51            60  3C 00036          MOVZWL  (R0), R1                ; 2423
                                51        04  A0  C0 00039          ADDL2   4(R0), R1               ;
                                61            20  90 0003D          MOVB    #32, (R1)               ;
                                              60  B6 00040          INCW    (R0)                    ; 2424
                                57        08  AE  3C 00042 3$:      MOVZWL  NUMDESC, R7             ; 2427
                                56        08  AC  D0 00046          MOVL    LINEDESC, R6            ; 2428
                                50            66  3C 0004A          MOVZWL  (R6), R0                ;
                                50        04  A6  C0 0004D          ADDL2   4(R6), R0               ;
                          60        0C    BE      57  28 00051          MOVC3   R7, @NUMDESC+4, (R0)    ;
                                50            66  3C 00056          MOVZWL  (R6), R0                ; 2429
                                50        04  A6  C0 00059          ADDL2   4(R6), R0               ;
                                6740          20  90 0005D          MOVB    #32, (R7)[R0]           ; 2430
                                50            66  3C 00061          MOVZWL  (R6), R0                ; 2431
                                51        01  A740  9E 00064          MOVAB   1(R7)[R0], R1           ; 2432
                                66            51  B0 00069          MOVW    R1, (R6)                ;
                                50            01  D0 0006C          MOVL    #1, R0                  ; 2434
                                                  04 0006F          RET                             ; 2435

; Routine Size:  112 bytes,    Routine Base:  $CODE$ + 1173
```

```
;  2158          2436  1 GLOBAL ROUTINE init_hex_octal =
;  2159          2437  1 BEGIN
;  2160          2438  1
;  2161          2439  1 !++
;  2162          2440  1 !
;  2163          2441  1 ! FUNCTIONAL DESCRIPTION:
;  2164          2442  1 !
;  2165          2443  1 !       Prepare for hex or octal output by building the required FAO
;  2166          2444  1 !       descriptors and initializing some global variables.
;  2167          2445  1 !
;  2168          2446  1 ! INPUTS:
;  2169          2447  1 !
;  2170          2448  1 !       None
;  2171          2449  1 !
;  2172          2450  1 ! OUTPUTS:
;  2173          2451  1 !
;  2174          2452  1 !       None
;  2175          2453  1 !
;  2176          2454  1 ! ROUTINE VALUES:
;  2177          2455  1 !
;  2178          2456  1 !       Always true
;  2179          2457  1 !
;  2180          2458  1 !--
;  2181          2459  1
;  2182          2460  1 LOCAL
;  2183          2461  1     fulldesc : BBLOCK [dsc$c_s_bln],
;  2184          2462  1     offsetsize,
;  2185          2463  1     partdesc : BBLOCK [dsc$c_s_bln];
;  2186          2464  1
;  2187          2465  1 !
;  2188          2466  1 ! Calculate the size of each entry in a line.
;  2189          2467  1 !
;  2190          2468  1 IF .dif$gl_flags [dif$v_hex]
;  2191          2469  1    THEN offsetsize = 9
;  2192          2470  1    ELSE offsetsize = 12;
;  2193          2471  1
;  2194          2472  1 !
;  2195          2473  1 ! Calculate the number of longwords to output per line.
;  2196          2474  1 ! Force it to be a power of two.
;  2197          2475  1 !
;  2198          2476  1 dif$gl_entsperline = MAXU (
;  2199          2477  1     (.dif$gl_width - dif$c_linenum - 1) / (.offsetsize + dif$c_entrysize),
;  2200          2478  1     1);
;  2201          2479  1 IF (.dif$gl_entsperline AND (.dif$gl_entsperline-1)) NEQ 0
;  2202          2480  1 THEN
;  2203          2481  1     dif$gl_entsperline = .dif$gl_entsperline AND (.dif$gl_entsperline-1);
;  2204          2482  1
;  2205          2483  1 !
;  2206          2484  1 ! Calculate the line width.
;  2207          2485  1 !
;  2208          2486  1 dif$gl_dumpwidth = .dif$gl_entsperline * (.offsetsize + dif$c_entrysize) + 8;
;  2209          2487  1
;  2210          2488  1 !
;  2211          2489  1 ! Pick the appropriate FAO descriptors to use.
;  2212          2490  1 !
;  2213          2491  1 dif$gl_faofulldesc [dsc$w_length] = dif$c_maxfaosiz;
;  2214          2492  1 dif$gl_faofulldesc [dsc$a_pointer] = dif$gl_faofullbuf;
```

```
;  2215          2493  2 dif$gl_faopartdesc [dsc$w_length] = dif$c_maxfaosiz;
;  2216          2494  2 dif$gl_faopartdesc [dsc$a_pointer] = dif$gl_faopartbuf;
;  2217          2495  2
;  2218          2496  2 IF .dif$gl_flags [dif$v_hex]
;  2219          2497  2    THEN BEGIN
;  2220          2498  3         fulldesc [dsc$w_length] = .hexfull [0];
;  2221          2499  3         fulldesc [dsc$a_pointer] = hexfull [1];
;  2222          2500  3         partdesc [dsc$w_length] = .hexpart [0];
;  2223          2501  3         partdesc [dsc$a_pointer] = hexpart [1];
;  2224          2502  3         END
;  2225          2503  3    ELSE BEGIN
;  2226          2504  3         fulldesc [dsc$w_length] = .octfull [0];
;  2227          2505  3         fulldesc [dsc$a_pointer] = octfull [1];
;  2228          2506  3         partdesc [dsc$w_length] = .octpart [0];
;  2229          2507  3         partdesc [dsc$a_pointer] = octpart [1];
;  2230          2508  2         END;
;  2231          2509  2
;  2232          2510  2 SYS$FAO (fulldesc, dif$gl_faofulldesc, dif$gl_faofulldesc,
;  2233          2511  2          .dif$gl_entsperline, .dif$gl_entsperline * dif$c_entrysize);
;  2234          2512  2 SYS$FAO (partdesc, dif$gl_faopartdesc, dif$gl_faopartdesc,
;  2235          2513  2          .dif$gl_entsperline * dif$c_entrysize);
;  2236          2514  2
;  2237          2515  2 RETURN true;
;  2238          2516  1 END;


                                 01FC 00000          .ENTRY  INIT_HEX_OCTAL, Save R2,R3,R4,R5,R6,R7,R8    ; 2436
                          58 00000000G   00   9E 00002       MOVAB   SYS$FAO, R8                          ;
                          57 00000000G   00   9E 00009       MOVAB   DIF$GL_FAOPARTDESC, R7               ;
                          56 00000000G   00   9E 00010       MOVAB   DIF$GL_FAOFULLDESC, R6               ;
                          55 00000000G   00   9E 00017       MOVAB   DIF$GL_ENTSPERLINE, R5               ;
                          54 00000000'   EF   9E 0001E       MOVAB   HEXFULL, R4                          ;
                          5E             10   C2 00025       SUBL2   #16, SP                              ;
53 00000000G   00         01             01   EF 00028       EXTZV   #1, #1, DIF$GL_FLAGS, R3             ; 2468
                          05             53   E9 00031       BLBC    R3, 1$                               ;
                          50             09   D0 00034       MOVL    #9, OFFSETSIZE                       ; 2469
                                         03   11 00037       BRB     2$                                   ;
                          50             0C   D0 00039 1$:    MOVL    #12, OFFSETSIZE                      ; 2470
                51 00000000G   00        07   C3 0003C 2$:    SUBL3   #7, DIF$GL_WIDTH, R1                 ; 2477
                          50             04   C0 00044       ADDL2   #4, R0                               ;
                          51             50   C6 00047       DIVL2   R0, R1                               ;
                                         03   12 0004A       BNEQ    3$                                   ; 2476
                          51             01   D0 0004C       MOVL    #1, R1                               ;
                          65             51   D0 0004F 3$:    MOVL    R1, DIF$GL_ENTSPERLINE               ;
                          52       FF    A1   9E 00052       MOVAB   -1(R1), R2                           ; 2479
                          52             51   D3 00056       BITL    R1, R2                               ;
                                         06   13 00059       BEQL    4$                                   ;
                          51             52   D2 0005B       MCOML   R2, R1                               ; 2481
                          65             51   CA 0005E       BICL2   R1, DIF$GL_ENTSPERLINE               ;
                          52             65   D0 00061 4$:    MOVL    DIF$GL_ENTSPERLINE, R2               ; 2486
                          50             52   C4 00064       MULL2   R2, R0                               ;
                00000000G   00       08    A0   9E 00067       MOVAB   8(R0), DIF$GL_DUMPWIDTH              ;
                          66                  28   B0 0006F       MOVW    #40, DIF$GL_FAOFULLDESC              ; 2491
                       04    A6 00000000G   00   9E 00072       MOVAB   DIF$GL_FAOFULLBUF, DIF$GL_FAOFULLDESC+4  ; 2492
```

```
                    67 28 B0 0007A          MOVW    #40, DIF$GL_FAOPARTDESC                         ; 2493
              04    A7 00000000G  00 9E 0007D       MOVAB   DIF$GL_FAOPARTBUF, DIF$GL_FAOPARTDESC+4 ; 2494
                    14 53 E9 00085          BLBC    R3, 5$                                          ; 2496
              08    AE  64 9B 00088         MOVZBW  HEXFULL, FULLDESC                               ; 2498
              0C    AE      01  A4 9E 0008C         MOVAB   HEXFULL+1, FULLDESC+4                   ; 2499
                    6E      4C  A4 9B 00091         MOVZBW  HEXPART, PARTDESC                       ; 2500
              04    AE      4D  A4 9E 00095         MOVAB   HEXPART+1, PARTDESC+4                   ; 2501
                    15 11 0009A             BRB     6$                                              ; 2496
              08    AE      6C  A4 9B 0009C 5$:     MOVZBW  OCTFULL, FULLDESC                       ; 2504
              0C    AE      6D  A4 9E 000A1         MOVAB   OCTFULL+1, FULLDESC+4                   ; 2505
                    6E    00B8  C4 9B 000A6         MOVZBW  OCTPART, PARTDESC                       ; 2506
              04    AE    00B9  C4 9E 000AB         MOVAB   OCTPART+1, PARTDESC+4                   ; 2507
        7E          52 02 78 000B1 6$:      ASHL    #2, R2, -(SP)                                   ; 2511
                    52 DD 000B5             PUSHL   R2                                              ;
                    56 DD 000B7             PUSHL   R6                                              ; 2510
                    56 DD 000B9             PUSHL   R6                                              ;
                    18  AE  9F 000BB        PUSHAB  FULLDESC                                        ;
                    68 05 FB 000BE          CALLS   #5, SYS$FAO                                     ;
        7E          65 02 78 000C1          ASHL    #2, DIF$GL_ENTSPERLINE, -(SP)                   ; 2513
                    57 DD 000C5             PUSHL   R7                                              ; 2512
                    57 DD 000C7             PUSHL   R7                                              ;
                    0C  AE  9F 000C9        PUSHAB  PARTDESC                                        ;
                    68 04 FB 000CC          CALLS   #4, SYS$FAO                                     ;
                    50 01 D0 000CF          MOVL    #1, R0                                          ; 2515
                       04 000D2             RET                                                     ; 2516

; Routine Size:  211 bytes,    Routine Base:  $CODE$ + 11E3
```

```
;  2240          2517  1 ROUTINE get_rfa_text (fdb, linedesc, width) =
;  2241          2518  2 BEGIN
;  2242          2519  2
;  2243          2520  2 !++
;  2244          2521  2 !
;  2245          2522  2 ! FUNCTIONAL DESCRIPTION:
;  2246          2523  2 !
;  2247          2524  2 !       Get a record from an input file using its RFA, and insert it in
;  2248          2525  2 !       the output line.
;  2249          2526  2 !
;  2250          2527  2 ! INPUTS:
;  2251          2528  2 !
;  2252          2529  2 !       fdb =       The address of the FDB for the input file.
;  2253          2530  2 !                   CURREC specifies the record to be read.
;  2254          2531  2 !
;  2255          2532  2 !       linedesc =  The address of a character string descriptor for the
;  2256          2533  2 !                   output line.
;  2257          2534  2 !
;  2258          2535  2 !       width =     The size of the input buffer specified by linedesc.
;  2259          2536  2 !
;  2260          2537  2 ! OUTPUTS:
;  2261          2538  2 !
;  2262          2539  2 !       The read record is appended to linedesc.
;  2263          2540  2 !
;  2264          2541  2 ! ROUTINE VALUES:
;  2265          2542  2 !
;  2266          2543  2 !       Always true
;  2267          2544  2 !
;  2268          2545  2 !--
;  2269          2546  2
;  2270          2547  2 MAP
;  2271          2548  2     fdb : REF BBLOCK,
;  2272          2549  2     linedesc : REF BBLOCK;
;  2273          2550  2
;  2274          2551  2 LOCAL
;  2275          2552  2     prefix_len,
;  2276          2553  2     rab : REF BBLOCK,
;  2277          2554  2     rdb : REF BBLOCK,
;  2278          2555  2     status;
;  2279          2556  2
;  2280          2557  2 rab = .fdb [fdb$l_rabptr];
;  2281          2558  2 rdb = .fdb [fdb$l_currec];
;  2282          2559  2
;  2283          2560  2 rab [rab$b_rac] = rab$c_rfa;                          ! Init RAB for RFA read
;  2284          2561  2 CH$MOVE (rfa$c_size, rdb [rdb$w_rfa], rab [rab$w_rfa]);
;  2285          2562  2
;  2286          2563  3 IF NOT (status = $GET (RAB = .rab))                   ! Get the record
;  2287          2564  2    THEN SIGNAL_STOP (dif$_readerr, 1, .fdb [fdb$l_fildesc],   ! If error, then signal
;  2288          2565  2                      .status, .rab [rab$l_stv]);
;  2289          2566  2
;  2290          2567  2 IF .rab [rab$w_rsz] GTR 0                            ! If record size greater than zero
;  2291          2568  3    THEN BEGIN
;  2292          2569  3         prefix_len = .linedesc [dsc$w_length];       ! Set amount of buffer already used
;  2293          2570  3         linedesc [dsc$w_length] = MIN0 (.width,      ! Calculate size of output string
;  2294          2571  3                   .linedesc [dsc$w_length] + .rab [rab$w_rsz]);
;  2295          2572  3         CH$MOVE (.linedesc [dsc$w_length] - .prefix_len,   ! Move record into buffer
;  2296          2573  3                  .rab [rab$l_rbf], .linedesc [dsc$a_pointer] + .prefix_len);
```

```
;  2297          2574  2          END;
;  2298          2575  2
;  2299          2576  2      !
;  2300          2577  2      ! After getting a record by RFA, we must reset the RMS pointers for
;  2301          2578  2      ! subsequent sequential reads to work.
;  2302          2579  2      !
;  2303          2580  2      rdb = .fdb [fdb$l_lastrec];
;  2304          2581  2      IF .rdb [rdb$v_eof]
;  2305          2582  2          THEN rdb = .fdb [fdb$l_lastrfa];
;  2306          2583  2      IF .rdb NEQ 0
;  2307          2584  3          THEN BEGIN
;  2308          2585  3               CH$MOVE (rfa$c_size, rdb [rdb$w_rfa], rab [rab$w_rfa]);
;  2309          2586  4               IF NOT (status = $FIND (RAB = .rab))
;  2310          2587  3                  THEN SIGNAL_STOP (dif$_readerr, 1, .fdb [fdb$l_fildesc],     ! If error, then signal
;  2311          2588  3                                   .status, .rab [rab$l_stv]);
;  2312          2589  3               rab [rab$b_rac] = rab$c_seq;                                    ! Reset RAB
;  2313          2590  4               IF NOT (status = $GET (RAB = .rab))
;  2314          2591  3                  THEN SIGNAL_STOP (dif$_readerr, 1, .fdb [fdb$l_fildesc],     ! If error, then signal
;  2315          2592  3                                   .status, .rab [rab$l_stv]);
;  2316          2593  2               END;
;  2317          2594  2
;  2318          2595  2      RETURN true;
;  2319          2596  1      END;


                                                        .EXTRN  SYS$GET, SYS$FIND

                                              0FFC 00000 GET_RFA_TEXT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 2517
                                  5B 00000000G   00  9E 00002           MOVAB   LIB$STOP, R11
                                  5A 00000000G   8F  D0 00009           MOVL    #DIF$_READERR, R10
                                  57       04    AC  D0 00010           MOVL    FDB, R7                     ; 2557
                                  56       30    A7  D0 00014           MOVL    48(R7), RAB
                                  58             67  D0 00018           MOVL    (R7), RDB                   ; 2558
                                        1E       A6  02 90 0001B       MOVB    #2, 30(RAB)                 ; 2560
                              10  A6        0C   A8  06 28 0001F       MOVC3   #6, 12(RDB), 16(RAB)        ; 2561
                                             56  DD 00025           PUSHL   RAB                         ; 2563
                                00000000G    00  01 FB 00027       CALLS   #1, SYS$GET
                                             59  50 D0 0002E           MOVL    R0, STATUS
                                             0F  59 E8 00031           BLBS    STATUS, 1$
                                           0C    A6  DD 00034           PUSHL   12(RAB)                     ; 2565
                                             59  DD 00037           PUSHL   STATUS
                                           38    A7  DD 00039           PUSHL   56(R7)                      ; 2564
                                             01  DD 0003C           PUSHL   #1
                                             5A  DD 0003E           PUSHL   R10
                                             6B  05 FB 00040       CALLS   #5, LIB$STOP
                                           22    A6  B5 00043 1$:      TSTW    34(RAB)                     ; 2567
                                             2D  13 00046           BEQL    3$
                                           50    08  AC  D0 00048           MOVL    LINEDESC, R0            ; 2569
                                             51      60  3C 0004C           MOVZWL  (R0), PREFIX_LEN
                                             53      60  3C 0004F           MOVZWL  (R0), R3                ; 2571
                                             52    22    A6  3C 00052           MOVZWL  34(RAB), R2
                                             53      52  C0 00056           ADDL2   R2, R3
                                             52    0C    AC  D0 00059           MOVL    WIDTH, R2
                                             53      52  D1 0005D           CMPL    R2, R3
                                                     03  1B 00060           BLEQU   2$                      ;
```

```
                                52              53  D0 00062          MOVL    R3, R2                                          ;
                                60              52  B0 00065 2$:      MOVW    R2, (R0)                                        ; 2570
                                53              60  3C 00068          MOVZWL  (R0), R3                                        ; 2572
                                53              51  C2 0006B          SUBL2   PREFIX_LEN, R3                                  ;
                    04 B041         28          53  28 0006E          MOVC3   R3, @40(RAB), @4(R0)[PREFIX_LEN]                ; 2573
                                58          08  A7  D0 00075 3$:      MOVL    8(R7), RDB                                      ; 2580
                          04        08          02  E1 00079          BBC     #2, 8(RDB), 4$                                  ; 2581
                                58          1C  A7  D0 0007E          MOVL    28(R7), RDB                                     ; 2582
                                45              13  00082 4$:         BEQL    6$                                              ; 2583
                    10    A6        0C          06  28 00084          MOVC3   #6, 12(RDB), 16(RAB)                            ; 2585
                                56              DD  0008A             PUSHL   RAB                                             ; 2586
                          00000000G             01  FB  0008C         CALLS   #1, SYS$FIND                                    ;
                                59              50  D0 00093          MOVL    R0, STATUS                                      ;
                                0F              59  E8 00096          BLBS    STATUS, 5$                                      ;
                                            0C  A6  DD 00099          PUSHL   12(RAB)                                         ; 2588
                                59              DD  0009C             PUSHL   STATUS                                          ;
                                            38  A7  DD 0009E          PUSHL   56(R7)                                          ; 2587
                                01              DD  000A1             PUSHL   #1                                              ;
                                5A              DD  000A3             PUSHL   R10                                             ;
                                6B              05  FB  000A5         CALLS   #5, LIB$STOP                                    ;
                                            1E  A6  94 000A8 5$:      CLRB    30(RAB)                                         ; 2589
                                56              DD  000AB             PUSHL   RAB                                             ; 2590
                          00000000G             01  FB  000AD         CALLS   #1, SYS$GET                                     ;
                                59              50  D0 000B4          MOVL    R0, STATUS                                      ;
                                0F              59  E8 000B7          BLBS    STATUS, 6$                                      ;
                                            0C  A6  DD 000BA          PUSHL   12(RAB)                                         ; 2592
                                59              DD  000BD             PUSHL   STATUS                                          ;
                                            38  A7  DD 000BF          PUSHL   56(R7)                                          ; 2591
                                01              DD  000C2             PUSHL   #1                                              ;
                                5A              DD  000C4             PUSHL   R10                                             ;
                                6B              05  FB  000C6         CALLS   #5, LIB$STOP                                    ;
                                50              01  D0 000C9 6$:      MOVL    #1, R0                                          ; 2595
                                                04  000CC             RET                                                     ; 2596

; Routine Size:  205 bytes,    Routine Base:  $CODE$ + 12B6

; 2320          2597  1
; 2321          2598  1 END                                                     ! Of module
; 2322          2599  0 ELUDOM


                                                                    .EXTRN  LIB$SIGNAL, LIB$STOP

;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                           401  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         4995  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                          562  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                              Library Statistics
;
;                                        --------- Symbols ---------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776        24         0       581          00:01.0




;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:OUTPUT/OBJ=OBJ$:OUTPUT MSRC$:OUTPUT/UPDATE=(ENH$:OUTPUT)

; Size:         4995 code + 963 data bytes
; Run Time:       01:21.8
; Elapsed Time:   02:46.0
; Lines/CPU Min:   1906
; Lexemes/CPU-Min: 20144
; Memory Used:  262 pages
; Compilation Complete
```

DIFMSG
LIS

DIRECTORY
LIS

MAIN
LIS

DIR

DIRECTORY
MAP

DIRECTDEF
REQ

OUTPUT
LIS

DISPLYDEF
SDL

DIRECTMSG
LIS